UCRL-MA-120465

INDUCT94: A Two-Dimensional Fluid Model of High Density Inductively Coupled Plasma Sources

**P. A. Vitello,[*] R. A. Stewart[†] D. B. Graves, [†]
E. F. Jaeger,[‡] and L. A. Berry[‡]**

March 1995

Lawrence
Livermore
National
Laboratory

[*] Lawrence Livermore National Laboratory, Livermore, CA 94550.
[†] Department of Chemical Engineering, UCB, Berkeley, CA 94720.
[‡] Oak Ridge National Laboratory, Oak Ridge, TN 38731.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

INDUCT94: A Two-Dimensional Fluid Model of High Density Inductively Coupled Plasma Sources

P. A. Vitello,[*] R. A. Stewart,[†] D. B. Graves,[†]
E. F. Jaeger,[‡] and L. A. Berry[‡]

[*]Lawrence Livermore National Laboratory, Livermore, CA 94550.
[†]Department of Chemical Engineering, UCB, Berkeley, CA 94720.
[‡]Oak Ridge National Laboratory, Oak Ridge, TN 38731.

Abstract

INDUCT94 is a Two-Dimensional numerical model of low pressure inductively coupled plasma sources based upon a fluid treatment of ions and electrons. Electron heating from external rf coils is calculated self-consistently by solving for the time-averaged rf electric field. We describe here the latest release of the INDUCT model, documenting in detail the code structure, variables, input, and output. We describe also the model equations and the algorithms used to solve them. A test case is presented, including output file data and post-processed graphics.

ACKNOWLEDGEMENT

One of the authors (R.A.S) gratefully acknowledges support from Sandia National Laboratory. Work by P.V. was performed under the auspices of the U.S. Department of Energy at the Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48.

1 Inductively Coupled Plasma Model

1.1 Introduction

New high plasma density sources with very good radial uniformity over large areas are being investigated and employed in ultra large scale integrated circuit manufacturing [1]. This is being driven by the trend towards large area, single wafer processing. The new sources share several characteristics desired for optimum performance. To achieve high ion flux (and hence large processing rates), the sources must be able to generate high plasma densities ($\gtrsim 10^{11}$ cm^{-3}) at low (< 30 mTorr) gas pressure. Low pressure is required to minimize ion collisions in the sheath above the etching surface and therefore maintain isotropy of the ion flux. An addition important characteristic is the low plasma potential ($<$ 10–30 V) and separate rf biasing of the substrate to control ion energies and reduce device damage. One of the most promising new plasma source is the Inductively Coupled Plasma (ICP) source, which has also been referred to in the literature by the names Radio Frequency Induction (RFI) and Transformer Coupled Plasma (TCP) source. One attractive feature of ICP's is the simplicity of their design. The ICP source may be considered as a multi-turn rf antenna coil coupled across a dielectric window to the plasma, with the plasma acting as a single-turn lossy conductor (see Figure 1). The ICP design is quite simple, with no external magnetic fields needed for efficient power coupling, and therefore no need for design and optimization of electro-magnetic or permanent magnet configurations. Separate rf coupling is applied via the substrate holder to modulate the extracted ion energy. Several recent studies have studies ICP's experimentally [2, 3, 4, 5, 6, 7, 8, 9]. Commercial ICP sources have recently become available [10, 11, 3], with typical operating conditions on input rf power of 200–1500 W, neutral gas pressures of 1–20 mTorr, and peak plasma densities of 10^{11}–10^{12} cm^{-3}.

Interest in plasma modeling has grown in recent years due to out improved understanding of the physics needed for such modeling, and due to the increased computational power available from current workstations which allow multi-dimensional time-dependent modeling to be practical. Modeling can provide understanding of the fundamental physics of plasmas. The aim of the model described here is to aid in the design and optimization of plasma reactors for integrated circuit manufacturing and other large area uses. High density reactors require at least two-dimensional time-dependent modeling with complex internal structures in order to accurately simulate power coupling and species transport. The model must simulate electron, ion, and neutral transport, include detailed volume and surface chemistry, and solve for both space charge, rf coil induced, and ac biased substrate rf generated electric fields. To address this modeling need we developed the INDUCT series of model, of which INDUCT94 is the most recent release. This paper documents INDUCT94 in detail, and is meant to be used as a user's guide to understanding this modeling tool.

1.2 Changes from INDUCT93

A new model of the INDUCT numerical inductively coupled plasma source modeling code has been prepared for general release. This new version, INDUCT94, contains many im-

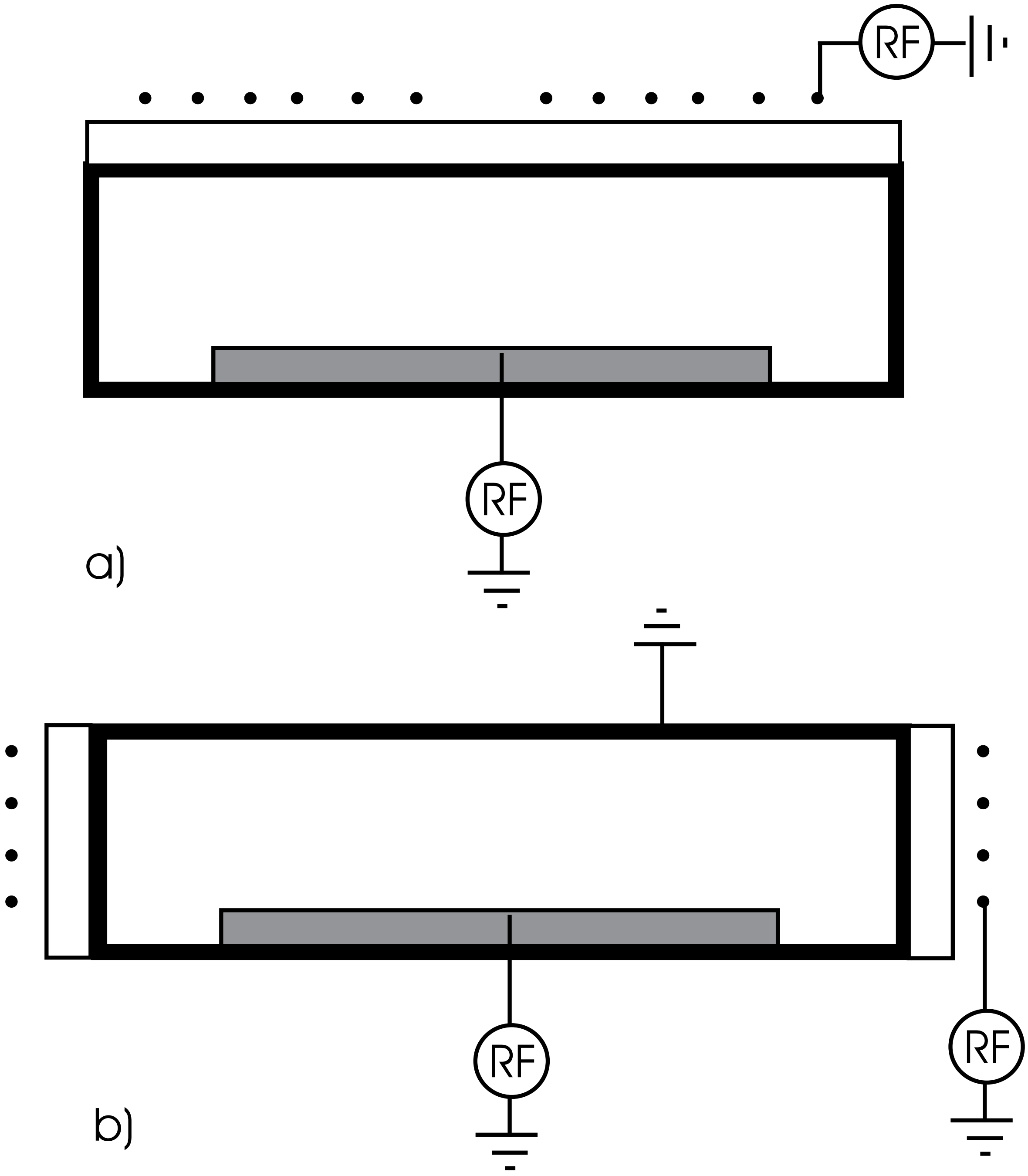


Figure 1: Typical geometry of an Inductively Coupled Plasma Source with substrate bias and two possible powered induction coil configurations: (a) planar coil; and (b) cylindrical coil.

provements over the previous release, INDUCT93. We give here a short list of improvements in INDUCT94, and a list of planned improvements to be included in INDUCT95, the next release version.

The initial release version of INDUCT, INDUCT93, was adequate for preliminary studies of inductively coupled plasma sources, but was applicable only in a narrow parameter regime because of its limited physical model. Additions to INDUCT93 have been made continuously since its release. We describe here he most significant improvements in INDUCT94.

The electron continuity equation is now solved using an implicit, conservative differencing scheme that allows time steps greater than the Courant time limit and the Dielectric Relaxation time scale. The Dielectric Relaxation time scale limit is overcome by using time advanced electron densities in Poissons equation calculated with the same algorithm used to solve the electron continuity equation. The electron continuity equation in conservative form also generates electron fluxes that are used in the electron temperature equation. To improve stability where steep density gradients occur, INDUCT94 allows the electron continuity equation to be solved using an upwind scheme. Generally, results differ only by a few per cent from the second order differencing scheme used previously, but now arbitrarily steep gradients can be handled. The upwind scheme is more stable in general and also allows larger time steps to be used. The second order scheme can still be used if desired. Self-consistency in the treatment of boundary conditions was enforced in INDUCT94. This improved the model accuracy and stability. The boundary conditions for ions are that the flux normal to a surface is continuous, the normal velocity is extrapolated, and the density is calculated as the flux divided by the density. This is for outward flow. If ions are found to be flowing into the plasma, then their flux, velocity, and density are assumed to be zero in the boundary cell just exterior to the plasma. For electrons, the outward flux at the boundary is given by thermal outward flow of electrons escaping across the sheath and an inward secondary emission flux. Multiple neutral and ion (both positive and negative) species are treated self-consistently in INDUCT94. The code was modifies so that an arbitrary number of species can be used, with chemical reactions coupling their evolution.

Atomic rates are now calculated using input rate tables. Atomic rates are a function of temperature and can have any dependence. Many new chemical reaction types have been added to the model. The chemistry model was generalized to allow two and three body reactions (with two or three reactants) and an arbitrary number of products. All rate information is now supplied via input, which allows for flexible chemistry modeling. Information input for each rate includes a rate type, the number of reactants and products, and an identifying string. The use of rate tables results in improved performance by removing the complex function evaluations used previously. A neutral chemistry model was added. Neutrals are assumed to be of uniform density with constant inflow. Both volume and surface chemistry are treated including wall recombination by ions to form neutrals, and surface chemistry reactions between neutrals. The neutral model is called every time step, but uses a larger time scale to accelerate convergence. rf biasing of the substrate has been added. The rf bias frequency can be set separately from the rf induction frequency. This allows an effective DC plasma potential to develop which modifies the energy of ions striking the substrate holder.

INDUCT93 was analyzed to improve computational performance. Effort was continuously made when new code elements or changes were made to maintain high computational efficiency. An improvement was make to limit plasma array computational loops to the region of the grid that bounded the plasma. The model mesh generally extended well beyond the plasma region to allow for boundary structures and rf coils. The restart scheme was rewritten to allow more flexible and predictable restarts. Units were standardized in input file. Comment lines in the code give units used. Out put data was reorganized to allow for arbitrary ion and neutral species. Wall chemistry data was also printed. Data for substrate rf period averaged variables was added. The rf induction field solver was replaced with an improved version from ORNL [13]. This new EM model uses a more realistic circuit model treating the rf coils as a transition line.

Improvements to INDUCT94 are already underway and will be available in the INDUCT95 release. What can be expected in the next release will include improved performance, more detailed output information, and a more complete physical model of an inductive discharge. The following enhancements are currently being considered. Not all of these features are likely to be available in the first INDUCT95 release.

The rf capacitive electric fields from the inductive coils will be incorporated into the fluid model to evolve the plasma. The chemistry model will be simplified so that a user can easily pick a gas mixture without needing to input extensive atomic physics details. The neutral model will be improved to treat 2D density, velocity, and temperature variations. An ion temperature equation will be added to the 2D fluid model. A hybrid electron and/or ion model will be incorporated. The hybrid model can be used to give ion and electron energy distribution functions for use in diagnostics and rate calculations. Incorporated an anomalous collision frequency into the conductivity used in the rf coil EM solver. Modifications to the rf coil EM solver will be made to improve speed and self-consistency. The fluid equations will be modified to include effects of strong magnetic fields.

1.3 Simulation Model

1.3.1 Fluid Equations

INDUCT94 solves a set of two-dimensional (cylindrically symmetric) time dependent fluid equations for electrons and ions self-consistently with Poisson's equation for the electric potential. In addition, rf inductive heating is calculated from a time-averaged solution of Maxwell's equations. Ions are assumed to be isothermal and near the neutral species temperature. Ion motion is governed by the equations of continuity and momentum conservation, which for ion species i are

$$\frac{\partial n_i}{\partial t} = -\vec{\nabla} \cdot n_i \vec{v}_i + \sum_{j=1}^{N_C} R_{ij}, \tag{1}$$

$$\frac{\partial n_i \vec{v}_i}{\partial t} = -\vec{\nabla} \cdot (n_i \vec{v}_i \vec{v}_i) + \frac{q_i n_i \vec{E}}{m_i} - \frac{1}{m_i} \vec{\nabla} n_i k T_i - \sum_{j=1}^{N_N} n_i \vec{v}_i \nu_{ij}. \tag{2}$$

Here n_i and $\vec{v}_i$ give the ion density and velocity, R_{ij} gives the chemical reaction rates leading to changes in the ion density, ν_{ij} is the ion neutral collision frequency, T_i is the ion temperature, m_i is the ion mass, q_i is the ion charge, and $\vec{E}$ is the electric field. The sums run over the total number of chemical reactions, N_C, and the total number of neutral species, N_N. The rates R_{ij} are evaluated using a table look-up from data read in during the initialization of the model. As described below, quite complex chemical interactions can be treated including ionization, attachment, and recombination. The ion-neutral collision frequency is calculated from the corresponding cross-section using

$$\nu_{ij} = \sigma_{ij} \bar{v}_i n_j, \tag{3}$$

where σ_{ij} is the ion-neutral cross-section between ion species i and neutral species j, n_j is the neutral density, and

$$\bar{v}_i = \left(\frac{8kT_i}{\pi m_i} + \vec{v}_i \cdot \vec{v}_i \right)^{1/2} \tag{4}$$

is the relative velocity between ions and neutrals. The cross-sections are assumed to be constant and are read in with the model input data.

The electron fluid model consists of the electron continuity and energy balance equations

$$\frac{\partial n_e}{\partial t} = -\vec{\nabla} \cdot \vec{\Gamma}_e + \sum_{j=1}^{N_C} R_{ej}, \tag{5}$$

$$\frac{\partial W_e}{\partial t} = -\vec{\nabla} \cdot \vec{Q} - e\vec{\Gamma}_e \cdot \vec{E} + P_{\text{ind}} - P_{coll}, \tag{6}$$

where

$$\vec{\Gamma}_e = -n_e \mu_e \vec{E} - \frac{1}{m_e \nu_N} \vec{\nabla} n_e k T_e, \tag{7}$$

is the electron flux in the "drift–diffusion" approximation, and

$$\vec{Q} = \frac{5}{2} \Gamma_e k T_e - \frac{5}{2} \frac{n_e k T_e}{m_e \nu_N} \cdot \nabla (kT_e) \tag{8}$$

is the electron energy flux. The electron density is given by n_e, $W_e = 3n_e k T_e/2$ is the electron thermal energy, m_e is the electron mass, μ_e is the electron mobility, ν_{eN} is the total electron-neutral collision frequency (summed over all neutral species), $P_{\rm ind}$ is the time-averaged power per unit volume absorbed by the electrons due to the inductive rf fields, and P_{coll} is the energy loss per unit volume due to electron-neutral collisions. Due to the slower ion response time, the drift-diffusion velocity approximation gives a poor representation of the ion velocity. We therefore solve the ion momentum equation directly.

The advective and chemistry terms in the ion and electron equations are solved separately (successively) by time splitting. The advective part of the electron continuity equation is solved implicitly to allow time steps greater than the Courant time limit and the Dielectric Relaxation time scale. A choice exists between using a first order in space Upwind scheme or second order spatial scheme for the electron advection terms in the continuity equation. This is chosen by input. The Upwind scheme is less accurate, but more stable allowing larger time steps to be used. The electron temperature equation is also solved implicitly to allow for large time step stability for thermal conduction and advection. A second order spatial scheme is used for the electron temperature scheme. In time spliting the electron continuity and electron temperature equations, the temperature held fixed in the continuity equation and the density held fixed in the temperature equation. This allows for the most efficient solution of these equations. Each implicit equation is solved using a second order in time Alternating-Direction-Implicit scheme. Because of the high mass of the ions relative to the electrons, the ion velocities are orders of magnitude slower than the electron velocity. This leads to the ion dynamic time scales being much longer than those for electron evolution. Due to the much slower ion time scale, an explicit temporal differencing scheme is used in the ion equations. Upwind differencing for stability is used for the ion equations.

Neutral flow and chemistry is treated in INDUCT94 assuming constant total pressure and uniform spatial distribution. A density for each neutral species is maintained at each grid point, with volume changes due to chemistry calculated in the same manner as is done for electrons and ions. In determining changes in the mean neutral densities, the volume chemistry changes are summed and combined with changes due to flow input and surface chemistry changes. The surface chemistry makes use of total ion and neutral species wall currents. Upon modifying the total neutral densities for each species, the densities are scaled by a constant to restore the required constant pressure. The neutral model subroutine is called every timestep along with the other fluid routines, but is allowed to use a time step which is larger. The neutral time step used is determined from the main time step variable DT multiplied by the input acceleration parameter ACCEL. Details of the chemistry model are given below.

The boundary conditions for the ion equations assume continuous outward flux and an extrapolated velocity. If the flux is determined to be inward, then it is set zero along with the boundary velocity. For electrons the flux is calculated from an analytic form $F_e = 0.25 n_e v_s e^{\delta\phi/kT_e} - \sum_{i=1}^{N_I} \gamma_i F_i$, where F_e is the outward electron flux, $v_s = \sqrt{8kT_e/\pi m_e}$ is the electron sound speed, γ_i is the secondary emission coefficient, F_i is the ion flux, and $\delta\phi$ is the potential drop from the plasma to the boundary structure. If $\delta\phi$ is positive, a zero value is used. The values of γ_i depend upon ion species and boundary material. If F_i is inward directed, then a zero value for F_i is used. The electron flux is assumed to be constant into the boundary structure so that using an extrapolated velocity, the exterior boundary density can be calculated. The electron temperature equation boundary condition used assumes an outward energy flux $Q_e = 2\Gamma_e kT_e$.

1.3.2 Poisson's Equation

Space-charge electric fields are determined self-consistently using Poissons equation. The electric field is evaluated using the electron and ion densities calculated from the continuity equations. Its value is re-calculated each time step. Complex internal boundaries can be treated for all equations. The effect of rf biasing of the substrate holder is included in INDUCT94 through the adjustment of the substrate holder potential at each time step.

The electrostatic electric field is calculated through the solution of Poisson's equation

$$\vec{\nabla} \cdot \epsilon \vec{\nabla} \phi = - \left(\sum_{i=1}^{N_I} q_i n_i + q_e n_e \right), \tag{9}$$

where ϵ is the local dielectric constant,q_i is the ion charge, and q_e is the electron charge. Poisson's equation is solved separately from other equations each time step. The use of fixed densities in the space-charge source term was found to be undesirable as this leads to a Dielectric Relaxation instability unless time steps on the order of picoseconds are used for electron densities of the order of 10^{11}cm^{-3}. The Dielectric Relaxation time scale is the electrostatic shielding time scale. To avoid disastrous amplification of the electrostatic field we solve Poisson's equation at the future time level using the time advanced electron density

$$n_e^{n+1} = n_e^n + \delta t \frac{\partial n_e}{\partial t}, \tag{10}$$

$$= n_e^n - \vec{\nabla} \cdot \vec{n}_e^n \mu_e^n \vec{\nabla} \phi^{n+1} - \frac{1}{m_e \nu_N^n} \vec{\nabla} n_e^n k T_e^n \tag{11}$$

where the superscript n implies the current time level variables, and superscript $n{+}1$ signifies the future time level variables to be solved for. Due so their slower response to the electric field, the ion densities are treated explicitly and not time advanced. Only the potential ϕ (which comes from the drift velocity term) is actually evaluated at the future time level, allowing the solution for the potential to be evaluated separately from that of the other fluid variables. For large time steps this insures near ambipolar fields and for steady state reduces to the simple form of Poisson's equation.

Dielectric boundary conditions are treated by summing ion and electron currents to each surface cell and distributing the surface charge through the first volume cell within the dielectric. This maintains charge balance and give the correct feed-back to Poisson's equation to generate electric fields which result in a cancelation between the electron and ion surface currents at each point along the dielectric.

In calculating the effect of an rf bias to the substrate we make use of a simple circuit connecting the substrate to ground (see Figure 2). The corresponding circuit equation is

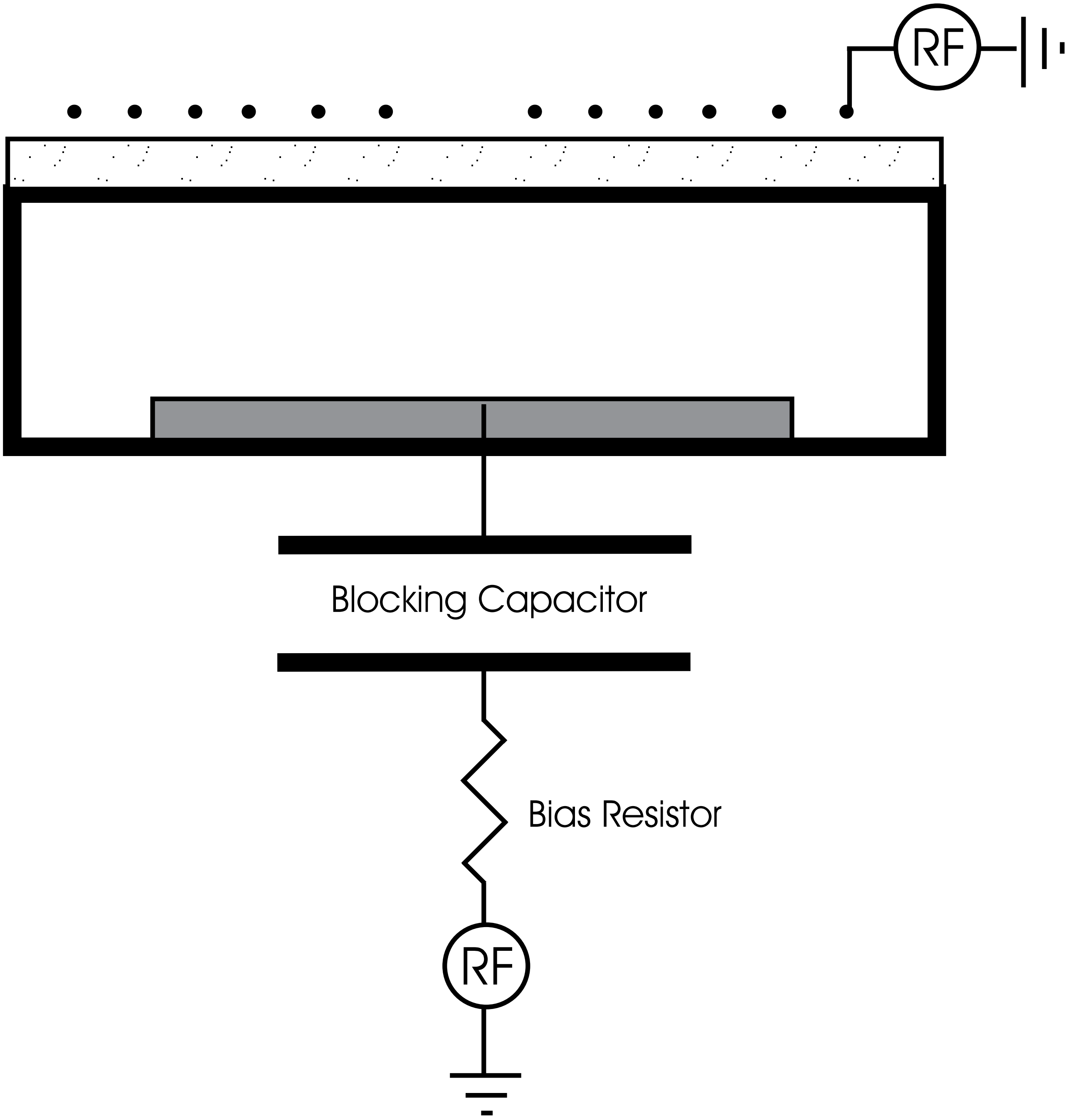


Figure 2: Circuit model for substrate rf biasing.

$$I_{\mathrm{rf}} = C_B \frac{\mathrm{d}}{\mathrm{d}t} \left(V_{\mathrm{rf}}(t) - \phi_{\mathrm{sub}} \right), \tag{12}$$

where I_{rf} is the circuit current, C_B is the blocking capacitor, $V_{\mathrm{rf}}(t)$ is the sinusoidal rf driving voltage, and ϕ_{sub} is the substrate holder voltage. The current I_{rf} can be related through charge conservation to the surface charge on the substrate holder, Q_{sub}, and the net plasma convection current into the substrate holder, I_p by

$$\frac{\mathrm{d}Q_{\mathrm{sub}}}{\mathrm{d}t} = I_{\mathrm{rf}} + I_p, \tag{13}$$

where Q_{sub} is calculated from the integral substrate holder surface charge density σ_{sub}

$$Q_{\mathrm{sub}} = \int_{\mathrm{sub}} \sigma_{\mathrm{sub}} \, dA. \tag{14}$$

The surface charge density in turn is calculated from Gauss's law applied to a small surface surrounding an element of the substrate holder which gives

$$\left(\epsilon_o \vec{E} - \epsilon_{\mathrm{sub}} \vec{E}_{\mathrm{sub}} \right) \cdot \hat{n} = \sigma_{\mathrm{sub}}, \tag{15}$$

where the terms with subscript *sub* are evaluated within the substrate. The unit vector $\hat{n}$ is normal to the substrate surface. Assuming a high conductivity for the substrate material we set $\vec{E}_{\mathrm{sub}} = 0$. Given the electric field at the current time level, $\vec{E}^n$, equation 15 is used to calculate the corresponding values of σ^n_{sub} from which the Q^n_{sub}. A simple first order explicit scheme is used to calculate I^n_{rf},

$$I^n_{\mathrm{rf}} = \frac{\left(\sigma^n_{\mathrm{sub}} - \sigma^{n-1}_{\mathrm{sub}} \right)}{\delta t} - I^n_p, \tag{16}$$

with δt being the time step. The substrate potential at the future time step, $\phi^{n+1}_{\mathrm{sub}}$ is then calculated using equation 16 as

$$\phi^{n+1}_{\mathrm{sub}} = \phi^n_{\mathrm{sub}} + V^{n+1}_{\mathrm{rf}} - V^n_{\mathrm{rf}} - \frac{\delta t}{C_B} I^n_{\mathrm{rf}}. \tag{17}$$

This scheme requires no iteration between the Poisson solver and the circuit equation and is accurate for time steps which resolve the substrate rf bias period.

1.3.3 rf Inductive Field Equations

The electromagnetic fields and rf heating arising from the inductive coils is calculated from a single-frequency approximation using a field solver supplied from ORNL. Resonance effects associated with long coils is treated. The coils are treated as circular current loops with rectangular cross section connected in series (see Figure 3). We consider an N loop antenna and let the coil index prescribe the order in which the coils are connected, with the first (or optionally the last) coil being driven directly by the generator with current I_{in} equal to the

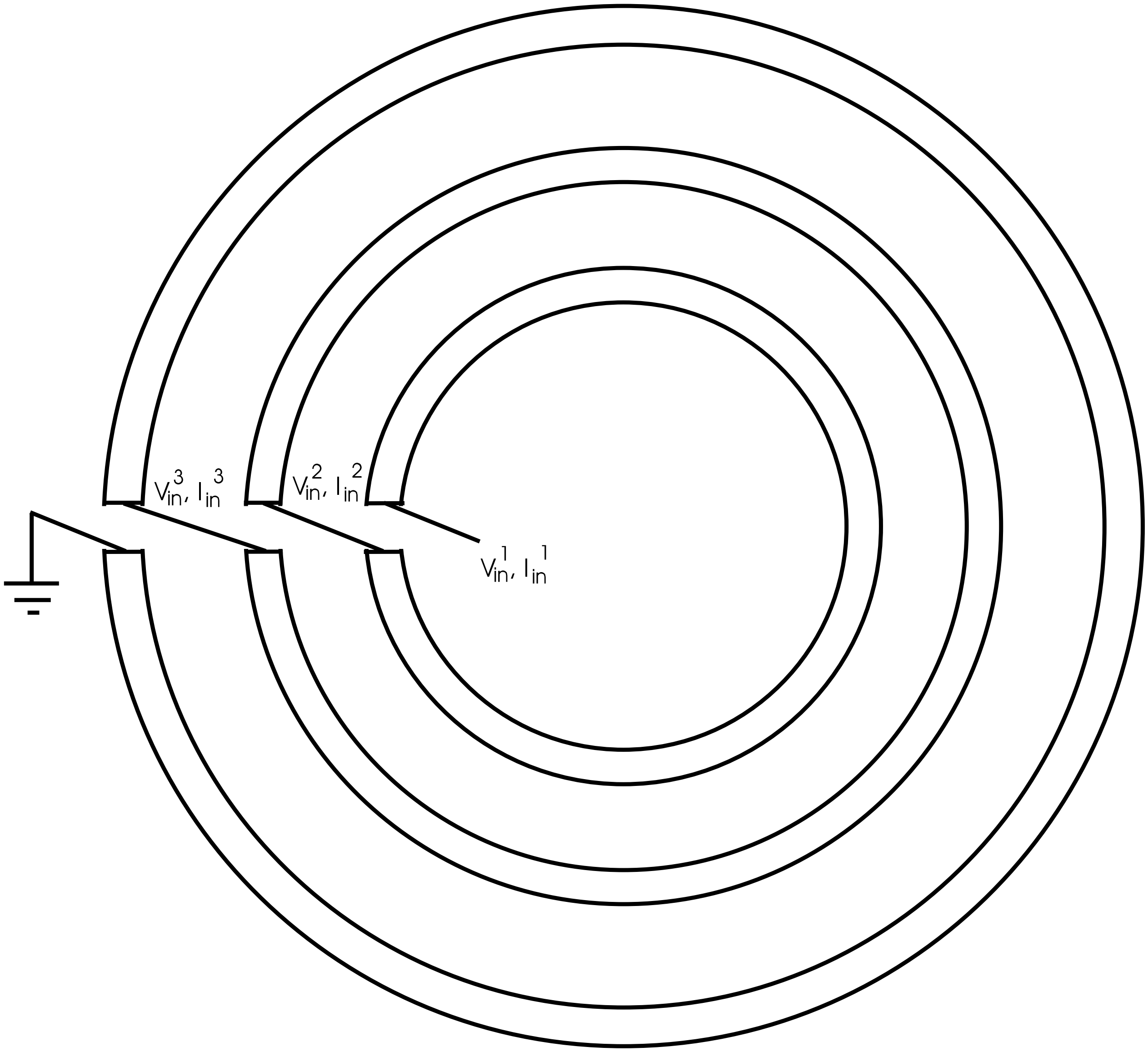


Figure 3: Geometry of rf inductive coils.

input variable IIN. The output coil is connected through a specified output impedance $Z_{\rm out}$ (given by input variable ZOUT) to ground. The average current in the ith coil is given by I_ℓ^i with input and output currents respectively of $I_{\rm in}^i$ and $I_{\rm out}^i$

As a result of the voltage needed to drive the inductive current, each coil will have an azimuthally varying potential. We let the average potential of the ith coil be given by $V_\mathcal{C}^i$, with $V_{\rm in}^i$ and $V_{\rm out}^i$ referring to the input and output potentials. All coil potentials are referenced to ground and defined by the loop integral

$$V^i(\theta) = -\oint_{2\pi}^{\theta} E_\theta^i d\theta - \sum_{j=i-1}^{N} \left[\int_{2\pi}^{\theta} E_\theta^j d\theta \right] + I_{\rm out}^N Z_{\rm out} \tag{18}$$

starting from the output end ($\theta = 2\pi$) of the Nth coil. The coil averaged quantities, I_ℓ^i and $V_\mathcal{C}^i$, are used as driving terms and boundary conditions respectively for the solution of the axisymmetric differential equations described below. Currently, these averages are estimated as $I_\ell^i = \left(I_{\rm in}^i + I_{\rm out}^i\right)/2$ and $V_\mathcal{C}^i = \left(V_{\rm in}^i + V_{\rm out}^i\right)/2$. Alternate methods for determining I_ℓ^i and $V_\mathcal{C}^i$ based on lumped circuit and transmission line models have been developed for use in situations when the capacitive current is not small compared to the inductive current. The difference between $I_{\rm in}^i$ and $I_{\rm out}^i$ can be approximated by the capacitive current flowing from each coil (see Equation 26), while the difference between $V_{\rm in}^i$ and $V_{\rm out}^i$ is obtained from Equation 18.

The plasma response to the inductive rf electric fields is modeled by the cold plasma dielectric tensor $\mathbf{K}$. For single frequency $e^{-i\omega t}$ time dependence,

$$\mathbf{K} = \mathbf{I} + \frac{i}{\omega\epsilon_o}\boldsymbol{\sigma}_p, \tag{19}$$

where $\mathbf{I}$ is the unity tensor, $\boldsymbol{\sigma}_p$ is the cold plasma conductivity, $\omega = 2\pi f$, and f is the rf coil driving frequency. With no steady state magnetic field, the conductivity tensor is diagonal and is given by

$$\boldsymbol{\sigma}_p = \frac{\epsilon_o \omega_{pe}^2}{\nu_{eN} - i\omega}\mathbf{I}, \tag{20}$$

where $\nu_{eN} = \sum_{i=1}^{N_N} \nu_{ei}$ is the total electron-neutral collision frequency, and $\omega_{pe} = e^2 n_e/\epsilon_o m_e$ is the electron plasma frequency. Other materials, such as quartz liners or windows, are characterized by the appropriate dielectric constant determined by input. In the case of a metal, $\mathbf{K}$ is dominated by the large real conductivity.

The driving frequency is assumed to be sufficiently high that ion motion is not important in the electromagnetic model. The form for the Fourier transformed Maxwell equations we solve is given by

$$-\vec{\nabla} \times \vec{\nabla} \times \vec{\mathcal{E}} + \frac{\omega^2}{c^2}\mathbf{K} \cdot \vec{\mathcal{E}} = -i\omega\mu_o \vec{J}_{coil}, \tag{21}$$

where c is the speed of light in free space, $\vec{J}_{coil}$ is the Fourier transform of the applied antenna current at the coils, $\mathcal{E}$ is the Fourier transform of the rf electric field, and μ_o is the permeability of free space. Poisson's equation follows by taking the divergence of the above equation in combination with $\vec{\nabla} \cdot \vec{J} - i\omega n_e = 0$:

$$\vec{\nabla} \cdot \mathbf{K} \cdot \vec{\mathcal{E}} = \frac{n_{\rm ext}}{\epsilon_o}. \tag{22}$$

In Equation 22, only the electron density n_e appears as this relation involves the Fourier transform of the electric field and ion motion is ignored. For axisymmetric driving currents with only a j_{ext_θ} component, Equations 21 and 22 reduce to

$$\nabla^2 \mathcal{E}_\theta + \frac{\omega^2}{c^2} K \mathcal{E}_\theta = -i\omega\mu_o J_{\text{ext}_\theta}, \tag{23}$$

and

$$\vec{\nabla} \cdot \left(K \vec{\nabla} \Phi \right) = -n_{e_{ext}} / \epsilon_o, \tag{24}$$

where K is the diagonal component to the dielectric tensor $\mathbf{K}$, $\vec{\nabla}\Phi = -\left(\mathcal{E}_r, \mathcal{E}_z\right)$, and $n_{e_{ext}} = 0$ since we consider Φ as arising from the coil voltages alone.

Equations 23 and 24 are solved by conventional finite difference techniques for $\mathcal{E}_\theta$ and Φ respectively. The boundary conditions for $\mathcal{E}_\theta$ are $\mathcal{E}_\theta = 0$ on all conducting surfaces and, by symmetry, $\mathcal{E}_\theta = 0$ at $r = 0$. The driving current J_{ext_θ} is derived from the coil currents I^i_ℓ. For equation 24, we prescribe Φ on the metal surfaces and require that $\partial\phi/\partial r = 0$ at $r = 0$. The potential on the ith coil is given by V^i_C. For an axisymmetric system, the equations for $\mathcal{E}_\theta$ and Φ are decoupled and the boundary conditions can be specified independently.

The time averaged rf heating electron rate is given by $\text{Re}\vec{J}_p \cdot \text{Re}\vec{\mathcal{E}}$, where $\vec{J}_p = \sigma_p \vec{\mathcal{E}}$ is the plasma current. Performing the time average over a wave period gives

$$P_{\rm ind} = \frac{1}{2}\text{Re}\left(\vec{\mathcal{E}}^* \cdot \vec{J}_p\right) = \frac{1}{2}\text{re}\left(\sigma_p \vec{\mathcal{E}} \cdot \vec{\mathcal{E}}\right). \tag{25}$$

While the model based upon equations 23 and 24 is axisymmetric, in real coils there is a dependence upon angle θ, and capacitive coupling from the coils should be included. The component of $\vec{\mathcal{E}}$ normal to a conductor surface is proportional to the displacement current flowing from that portion of the conductor. For the complete coil we have

$$I^i_C = -i\omega\epsilon_o \oint_{A_i} \vec{\mathcal{E}} \cdot \hat{n} dA, \tag{26}$$

where the integral is over a surface which encloses the ith coil. The current I^i_C represents the capacitive current flowing from the ith coil. This current breaks the axisymmetry of the system. If, for example, the normal electric field were cylindrically symmetric, the capacitive current density leaving (or entering) the conductor would be uniform along the coil, and the conduction current would have a linear dependance upon angle θ.

While the change in current along a given turn cannot be represented by this symmetric model, the change in current from turn to turn can be. This is done in an iteration scheme as follows:

1. Use the average potential V_ℓ^i on the coils to solve equation 24 for $(\mathcal{E}_r, \mathcal{E}_z)$, and then solve equation 26 to determine the capacitive currents $I_\mathcal{C}^i$.
2. Impose conservation of current by requiring that the change in current from the ith to the $(i+1)$th coil be give by the capacitive current in the ith coil, so that $I_{\text{in}}^i = I_{\text{out}}^i + I_\mathcal{C}^i$.
3. Use the average currents I_ℓ^i to solve equation 23 for $\vec{\mathcal{E}}_\theta$.
4. Solve equation 25 for the power density and sum its value over the plasma volume. As the power density scales with input current squared, I_{in}^2, a desired total power absorption can be obtained by scaling I_{in} at this point.

Self-consistency is obtained when the specified inductive currents I_ℓ^i result in potentials $V_\mathcal{C}^i$ that are just those needed to produce the turn-to-turn variations in the inductive currents.

1.3.4 Chemistry Model

The charged plasma species are generated from and transported through a background of neutrals. The simulation currently uses a spacially uniform, but time evolving set of neutral densities. Volume rates are input during setup and used as a rate table. There are currently data sets available for INDUCT94 for argon and chlorine. The corresponding input files to be used are INDUCT94.AIN and INDUCT94.CIN. Note that the appropriate input file must be renamed INDUCT94.IN before running the model. The species included for the argon model are: electrons, Ar^+, and Ar. The chemistry rate files for argon are the following:

1. ioniz.k
2. ecitmeta.k
3. elastic.k
4. excit3d132.k
5. excit3d32.k
6. excit5d12.k
7. excit5s32.k

and should be placed in a subdirector named argon relative to where INDUCT94 is run. These rates correspond to

1. Ar Ionization: $e + Ar \longrightarrow Ar^+ + 2e$
2. Ar excitation to metastable: $e + Ar \longrightarrow Ar^* + e \longrightarrow Ar + e$
3. Ar elastic scattering: $e + Ar \longrightarrow e + Ar$

4. Ar excitation to 3d^1 3/2: e + Ar $\longrightarrow$ Ar* + e $\longrightarrow$ Ar + e
5. Ar excitation to 3d 3/2: e + Ar $\longrightarrow$ Ar* + e $\longrightarrow$ Ar + e
6. Ar excitation to 3d 1/2: e + Ar $\longrightarrow$ Ar* + e $\longrightarrow$ Ar + e
7. Ar excitation to 5s 3/2: e + Ar $\longrightarrow$ Ar* + e $\longrightarrow$ Ar + e

Where it is relevant, a value for the energy lost is contained in the rate table . For elastic scattering we use as the energy lost per scattering the value $\epsilon_{\text{elas}} = (2m_e/m_N)kT_e$, where m_N is the neutral species the electron is scattering with. The species included in the chlorine model are: electrons, Cl, Cl_2, Cl^-, Cl^+, and Cl_2^+. For chlorine, the chemistry rate files are the following:

1. Clion.k
2. Clmom.k
3. Cl3d.k
4. Cl4d.k
5. Cl4p.k
6. Cl4s.k
7. Cl5d.k
8. Cl5p.k
9. Cl5s.k
10. Cl6s.k
11. Cl2ion.k
12. Cl2mom.k
13. Cl2exc.k
14. Cl2vib.k
15. Cl2dis.k
16. Cl2att.k
17. Cl-Cl+.k
18. Cl-Cl2+.k

19. Cl-det.k

and should be placed in a subdirector named chlorine relative to where INDUCT94 is run. These rates correspond to

1. Cl Ionization e + Cl $\longrightarrow$ Cl^+ + 2e

2. Cl Momentum Transfer e + Cl $\longrightarrow$ Cl + e

3. Cl 3d Excitation: e + Cl $\longrightarrow$ Cl^* + e $\longrightarrow$ Cl + e

4. Cl 4d Excitation: e + Cl $\longrightarrow$ Cl^* + e $\longrightarrow$ Cl + e

5. Cl 4p Excitation: e + Cl $\longrightarrow$ Cl^* + e $\longrightarrow$ Cl + e

6. Cl 4s Excitation: e + Cl $\longrightarrow$ Cl^* + e $\longrightarrow$ Cl + e

7. Cl 5d Excitation: e + Cl $\longrightarrow$ Cl^* + e $\longrightarrow$ Cl + e

8. Cl 5p Excitation: e + Cl $\longrightarrow$ Cl^* + e $\longrightarrow$ Cl + e

9. Cl 5s Excitation: e + Cl $\longrightarrow$ Cl^* + e $\longrightarrow$ Cl + e

10. Cl 6s Excitation: e + Cl $\longrightarrow$ Cl^* + e $\longrightarrow$ Cl + e

11. Cl_2 Ionization: e + Cl_2 $\longrightarrow$ Cl_2 + 2e

12. Cl_2 Momentum Transfer: e + Cl_2 $\longrightarrow$ Cl_2 + e

13. Cl_2 Electronic excitation: e + Cl_2 $\longrightarrow$ Cl_2^* + e $\longrightarrow$ Cl_2 +e

14. Cl_2 Vibrational excitation: e + Cl_2 $\longrightarrow$ Cl_2v + e $\longrightarrow$ Cl_2 + e

15. Cl_2 Dissociation: e + Cl_2 $\longrightarrow$ 2Cl + e

16. Cl_2 Dissociative attachment: e + Cl_2 $\longrightarrow$ Cl^-1 + Cl

17. Atomic neutralization: Cl^+ + Cl^- $\longrightarrow$ 2Cl

18. Molecular neutralization: Cl_2^+ + Cl^- $\longrightarrow$ 3Cl

19. Cl^-1 Detachment: e + Cl^- $\longrightarrow$ Cl + 2e

Data sets for CF_4 and other gas mixtures are being developed. We are also working to expand the existing models by adding metastable neutral species to the argon model and additional chlorine reactions to the chlorine model.

2 INDUCT94 Coding

2.1 Files

The distribution of INDUCT94 consists of the following text files:

1. EM94.INC
2. INDUCT94.F
3. INDUCT94.H
4. INDUCT94.AIN
5. INDUCT94.CIN
6. INDUCT94.INC
7. INDUCT94.PS
8. SID94.INC
9. WAVE.PLT

Files with the extension .INC are FORTRAN files containing common block variables. These files are "included" by the compiler into the FORTRAN source file INDUCT94.F. Files ending with the extension .H are FORTRAN files which contain FORTRAN parameter variables. Such files are also "included" by the compiler into the source file INDUCT94.F. The files with extension .AIN and .CIN contain input data used by INDUCT94. Before running INDUCT94, one of these two input files must be renamed to INDUCT94.IN. The test problem described below makes use of the INDUCT94.AIN. In "including" source files, the file INDUCT94.H is read first to provide parameter definitions. INDUCT94.H also contains an IMPLICIT NONE statement which specifies that all variable must be defined before usage. The file INDUCT94.PS is a Post-Script file containing this documentation.

2.1.1 EM94.INC

This file contains common blocks and variable definitions used to communicate with the rf coil electro-magnetic (EM) field solver routines. The variable definitions are reproduced here for reference

```
 Common Block Variable Descriptions
 (Units of input variables used in EMSOURCE subroutine reflect their usage
  in the subroutine (MKS), not their usage in the namelist (mostly CGS).
  Unit conversion of input variables takes place in the SETUP subroutine)
**************************************************************************
 integer icdim          Maximum number of coils.
integer icheck         If(icheck.eq.0) no check on em equation solution.
                       If(icheck.eq.1) em equations are checked.
                          (not used at present)
integer imat(nrdim,nzdim)
                       em routines material array.
                               imat(n,m)=0:   plasma
                               imat(n,m)=1:   coil
                               imat(n,m)=2:   quartz
                               imat(n,m)=3:   vacuum
                               imat(n,m)=4:   silicon
                               imat(n,m)=5:   cathode
                               imat(n,m)=6:   boundary (ground)
integer imax           Maximum number of points for em sheath data.
                          (not used at present)
integer ldamax         6*nzdim + 7. First dimension of em matrix adb.
integer mdiag          Location of z diagnostic print in em routine.
                          (not used at present)
integer ndiag          Location of z diagnostic print in em routine.
                          (not used at present)
integer niter          Number of iterations to do for coil current.
integer nmatmx         2*nrdim*nzdim. Second dimension of em matrix adb.
integer nrdim          Maximum number of r cell boundaries in em routines.
integer nzdim          Maximum number of z cell boundaries in em routines.

real*8  epsit          Fractional accuracy limit on current iteration.
real*8  radius1        Inner radius of bell jar.  (m)
real*8  radius2        Outer radius of bell jar.  (m)
real*8  rcath          Radius of em cathode in r.  (m)
real*8  rcenter        Radius at which bell jar curvature begins. (m)
real*8  rdop           Radius of doped silicon in em routines.  (m)
real*8  rwin1          Inner boundary of em window in r.  (m)
real*8  rwin2          Outer boundary of em window in r.  (m)
real*8  sigdop         em conductivity in doped silicon.
                          (not used at present)
real*8  xkwin          Dielectric constant for em window.
real*8  xncath         Density of em cathode metal.  (cm-3)
real*8  zcath          Height of em cathode in z.  (m)
real*8  zcenter        Axial location at which bell jar curvature
                          begins. (m)
real*8  zdop           Lower boundary of doped silicon in z for em
                          routines. (m)
real*8  zwin1          Lower boundary of em window in z.  (m)
```

```
 real*8  zwin2          Upper boundary of em window in z.  (m)
```

2.1.2 INDUCT94.F

FORTRAN source code for INDUCT94.

2.1.3 INDUCT94.H

FORTRAN parameter variables used by INDUCT94 in the plasma model. There definitions are give within the file and are reproduced here.

```
 Parameter Descriptions
**************************************************************************

 integer mbound    Maximum number of mask boundary cells allowed.
 integer mchem     Maximum number of chemical reactions.
 integer mmask     Maximum number of masked structures.
 integer mr        Maximum number of r grid cells.
 integer mrp1      mr + 1.
 integer mrp2      mr + 2.
 integer mrepd     Maximum number of reactants or products per
                        chemical reaction.
 integer mrfstep   Maximum number of rf time steps over which rf
                        diagnostic data is saved.
 integer mspec     Maximum number of chemical species.
 integer mvalue    Maximum number of rate constant values.
 integer mz        Maximum number of z grid cells.
 integer mzp1      mz + 1.
 integer mzp2      mz + 2.
```

2.1.4 INDUCT94.IN

Input file for INDUCT94. Input data for the EM field solver is found in this file as well as input for the general plasma model routines. Variables as formated using NAMELIST. Input variables are defined in file INDUCT94.F after the NAMELIST statement in subroutine OUTPUT and are reproduced here.

```
 <<  input parameters mainly for plasma routines.

    rmax     -Total width in r direction.  (cm)
    rsheath  -Thickness of region in r about rmax surface where non-linear
              gridding may occur.  If rsheath <= 0 use uniform r grid. (cm)
              (Use with caution!)
    zmax     -Total height in z direction.  (cm)
```

zsheath -Thickness of region in z about 0 and zmax surfaces where non-linear gridding may occur. If zsheath <= 0 use uniform z grid. (cm) (Use with caution!)

nr -Total number of r grid cells.

nri -Number of r grid cells in rsheath surface region. (Use with caution!)

nz -Total umber of z grid cells.

nzi -Number of z grid cells in zsheath surface region. (Use with caution!)

phir -Reference potential. Potential added to all applied structures. Generally phir = 0 used. (V)

den0 -Initial peak ion & electron density. (cm-3)

mstep -Maximum number of time steps.

mstepp -Maximum number of time steps between data dumps from output routine.

msteppr -Maximum number of steps between calls to iheat subroutine for inductive rf heating.

dt0 -Initial time step. (s)

mt -Maximum time. (s)

mtp -Maximum time between data dumps from output routine. (s)

mtpr -Maximum time between calls to iheat subroutine for inductive rf heating.

iorder -Determines spatial differencing scheme order for the electron continuity equation. iorder=1 => upwind differencing, iorder=2 => second order differencing.

cour -Courant condition limit. If acbias is zero, then cour usually determines the time step after the code has increased the time step from the initial value dt0. A value of cour = 1 is conservative. Values of cour from 5--10 should work well. The electron and ion densities will get noisy for too large values of cour.

dtefactor-Extrapolation factor for time step. The new time step may be no larger than dtefactor larger than the old time step. A value of dtefactor close to unity allows a smooth start up from the initial time step dt0.

dtifactor-Chemistry time step factor. Gives approimate maximum change in density allowed per time step due to chemistry.

secondary-Secondary emission coefficients. First index is over species. Only entries for ions have meaning. Second index is for material (cathode, metal wall, dielectric wall).

phiacc -Fractional change convergence limit for phi for SIP poisson solver.

```
phiacco  -Fractional change convergence limit for phi for SOR poisson
          solver.
aparam   -SIP acceleration parameter.
conres   -SIP residual error limit factor.
istart   -Restart dump to use, if zero start calculation from t = 0.
itmaxsip -Maximum number of SIP iterations for Poisson solution.
itmaxsor -Maximum number of SOR iterations for Poisson solution.
msip     -Use SIP iteration every msip time steps after starting with
          the SIP scheme.  During the initial transient period for a
          simulation it is best to use msip = 1--2.  For no ac bias,
          larger values of msip can be used upon restarting the
          model when the time variation of parameters has slowed.
nspec    -Total umber of fluid species including electrons.
negspec  -Number of negative ion species.
neutspec -Number of neutral species.
spect    -Species type.  Values are 1 for electron, 2 for ion, -2 for
          negative ion, and 3 for neutral.
specl    -String describing species.
specq    -Charge of ion and neutral fluid species.
specm    -Mass of ion and neutral fluid species in proton mass.  This
          variable is multiplied by the proton mass during
          initiallization so that its internal value gives the
          species mass in grams.  The first index is filled with the
          electron mass in subroutine SETUP.
nchem    -Total number of chemical reactions.
ic       -Reactant and product species index list for each reaction.
          The value of this variable tells which species are the
          reactants and which are the products for a given reaction.
          The chemical reaction being considered is determined by the
          second index.  The first index gives which species are
          involved.  The following table gives allowed reactions.
          Ratet gives the chemical reaction type.  These interactions
          can involve electrons (e), ions (i), and neutrals (n).  Ions
          and neutrals are designated as reactants (r) or products (p)
          to signify that the product ion or neutral species may be
          different from the reactant species.  For reactions where the
          product ion or neutral is the same as the reactant ion or
          neutral, the product species is labeled (r) to show that it
          is the same species as the reactant.  Entries marked with ...
          imply than this reaction can have an arbitrary number of
          product neutral species.
           ratet  ic(1,i)  ic(2,i)  ic(3,i)  ic(4,i)  ic(5,i)
```

```
                 1      e       n(r)     i(p)      e        e
                 2      e       n(r)     n(r)      e
                 3      e       n(r)     n(r)      e
                 4      e       n(r)     i(p)     n(p)     ...
                 5      e       n(r)      e       n(p)     n(p) ...
                 6      e       n(r)     i(p)     i(p)      e
                 7      e       i(r)     n(p)      e        e
                 8     i(r)     i(r)     n(p)     n(p)
                 9     i(r)     i(r)     n(p)     n(p)     ...
                10      e       n(r)      e       i(p)     n(p) ...
                11      e       i(r)     i(p)      e
                11      e       n(r)     n(p)      e
ratel    -String describing rates.
ratet    -Rate type.
        ( 1) Electron impact ionization.                     (irate)
        ( 2) Electron/neutral elastic.                       (erate)
        ( 3) Excitation (with radiative decay).              (xrrate)
        ( 4) Dissassociative attachment.                     (darate)
        ( 5) Dissassociation.                                (dirate)
        ( 6) Dissassociation with ionization and attachment. (diarate)
        ( 7) Detachment.                                     (derate)
        ( 8) Atomic neutralization.                          (anrate)
        ( 9) Molecular neutralization.                       (mnrate)
        (10) Dissassociation with ionization.                (diirate)
        (11) Excitation or dexcitation.                      (xrate)
fracinit -Initial fractional loading of ions or neutrals.  There is a
          separate accounting for electrons and negative ions, positive
          ions and neutrals.  The fracinit values for positive ions and
          neutrals should separately sum to unity.  The fracinit values
          for negative ions should sum to less than unity, with the
          difference from unity assigned to electrons.  The plasma
          is initially neutral with the local number density of positive
          and negative charge species equal.
csectin  -Ion-neutral collision cross section.  The first index gives
          the ion species.  The second index gives the neutral. (cm+2)
tempi0   -Initial value for ion temperature.  Currently the ion
          temperature is held fixed in space and time.  (V)
tempe0   -Initial value for global electron temperature.  (V)
localte  -If .true. use spatially varying electron temperature model.
          Otherwise use a global spatially uniform model for the
          electron temperature.
localti  -If .true. use spatially varying ion temperature model.
```

```
          (Use .false. only for current model as ion temperature
          equation does not currently contain all interaction terms!)
uniform  -If .true. use uniform power deposition model (inductive rf em
          solver not called).  If .false. then spatially varying
          inductive rf power calculated through calls to iheat routine.
multidump-If .true. drop multiple sets of dump files, otherwise
          dump files are overwritten.  As restarts are make from the
          dump files, if multidump is .false., then the code can be
          restarted only from its termination point and not at an
          earlier time.
mcycle   -Maximum number of substrate rf cycles.  The code will
          terminate when mcycle rf cycles have occurred.
mcyclep  -Maximum number of substrate rf cycles between data dumps.
mrf      -Number of time steps per substrate rf cycle.  If acbias is
          not zero, then mrf determines the time step.
nrfstepmod
         -rf data in induct94.rf file saved every nrfstepmod time steps.
```

<< input parameters for neutral species.

```
pressure -Total neutral gas pressure.  (mTorr)
nchems   -Number of surface chemical reactions.
nreacts  -Number of reactants per surface chemical reaction.
nprods   -Number of products per surface chemical reaction.
ics      -Reactant & product species index list for each
          surface reaction.  Its use is similar to the that for ic.
ratest   -Surface rate type.
ratesl   -Surface rate label string.
tempn0   -Initial uniform neutral temperature.  (V)
flow     -Neutral feed gas flow rate.  (sccm)
gammar   -Neutral wall recombination coefficient for ions.
gammai   -Wall recombination coefficient for ion corresponding
          to a given neutral species.
gammac   -Wall chemistry reaction coefficient.
accel    -Neutral timestep acceleration.  The neutral density evolves
          with a time step that is accel times larger than the
          plasma time step.
freact   -Rate constant lookup table file names.
```

<< input parameters mainly for inductive rf EM solver and substrate
rf bias.

```
lbias    -If .true., then substrate rf bias turned on.
dcbias   -Substrate DC bias voltage.  (V)
acbias   -Substrate AC bias voltage.  (V)
biasfreq -rf frequency of substrate.  (Hertz)
biascapac-Capacitor in series with substrate rf bias voltage source.
          (Farads)
biasresistance
         -Resistor in series with substrate rf bias voltage source.
          (Ohm)
nrem     -Number of radial mesh (cell boundary) points for em solver.
nzem     -Number of axial mesh (cell boundary) points for em solver.
ncoil    -Number of coils.
r1coil   -Inner coil radii array. cm on input, m in em routines.
r2coil   -Outer coil radii array. cm on input, m in em routines.
z1coil   -Lower coil boundary array. cm on input, m in em routines.
z2coil   -Upper coil boundary array. cm on input, m in em routines.
ifeed    -If ifeed = 1 voltage is applied to first coil,
          else if ifeed = 2 voltage is applied to last coil.
coilfreq -Coil RF frequency.  (Hertz)
coilcapcoup
         -True if including capacitive coupling from coils to plasma.
          (not used at present)
prf      -If(uniform .eqv. .true.) then prf is the inductive power
          absorbed in the plasma, and the power is uniformly deposited.
          If(uniform .eqv. .false.) then
             if(prf.ne.0.) prf is the power absorbed non-uniformly
             and the current iin is scaled to be consistent with prf.
             if(prf.eq.0.) iin remains fixed at initial guess and
             the inductive power is determined by iin.  (W)
iin      -Initial guess for complex current entering the driven
             coil.  (A)
zout     -Output impedance at coil termination.  (ohms)
niter    -Number of iterations on coil current.
epsit    -The accuracy limit on current iteration.
zwin1    -Lower boundary of top window.  cm on input, m in em routines.
zwin2    -Upper boundary of top window.  cm on input, m in em routines.
rwin1    -Inner boundary of side window. cm on input, m in em routines.
rwin2    -Outer boundary of side window. cm on input, m in em routines.
rcenter  -Radius at which center jar curvature begins.  cm on input,
             m in em routines.
zcenter  -Axial location at which center jar curvature begins.  cm on
```

```
             input m in em routines.
radius1  -Inner radius of bell jar.  cm on input, m in em routines.
radius2  -Outer radius of bell jar.  cm on input, m in em routines.
rblock   -Inner radius for blocking dielectric.  This region runs in r
             from rblock to one cell inside of rmax.  (cm).
zblock   -Height for blocking dielectric.  This region runs in r from
          one cell above z = 0 to zblock.  (cm)
xkwin    -Dielectric constant for window.
rcath    -Radius of cathode. cm on input, m in em routines.
zcath    -Height of cathode. cm on input, m in em routines.
xncath   -Density of cathode metal.  (cm-3)
```

2.1.5 INDUCT94.INC

Contains main common blocks and variable definitions used by the INDUCT94 simulation modeling routines. Variable definitions are reproduced here for reference

```
 Common Block Variable Descriptions

 (Units of input variables used in EMSOURCE subroutine reflect their usage
  in the subroutine (MKS), not their usage in the namelist (mostly CGS).
  Unit conversion of input variables takes place in the SETUP subroutine.)
***************************************************************************

 logical coilcapcoup   .true. if including capacitive coupling from coils
                          to plasma.
 logical done          .true. when simulation is finished, otherwise .false.
 logical lbias         .true. input variable determininf if ac bias to
                          be used.
 logical localte       .true. for a locally (spatially) varying electron
                          temperature.  If .false., then use global
                          temperature model.
 logical localti       .true. for a locally (spatially) varying ion
                          temperature.  If .false., ion temperature is held
                          at initial value.
 logical lzerone       Determines model used to calculate extrapolated
                          electron density across boundaries.  If .true.,
                          use zero electron exterior density boundary
                          unless this implies an inward electron flux.  If
                          .false., use flux boundary condition:
                          Fe = 0.25*ne*vs*exp(delta phi/Te) - secondary*Fi.
                          Fe is the electron boundary flux, Fi is the
                          outward directed ion flux, ne is the electron
```

```
                         density, vs is the electron sound speed, Te is
                         the electron temperature, and delta phi is the
                         potential gradient from the plasma cell adjacent
                         to the boundary to the boundary structure.  If
                         delta phi is positive, a zero value is used.
logical multidump      .true. to save multiple sets of dump files, otherwise
                         dump files are overwritten.
logical uniform        .false. to invoke inductive rf em solver, otherwise
                         use uniform power deposition if .true.


character*3  ffile(18)Dump/restart file name prefixes.
character*5  fnfile(18)
                      Dump/restart file unit numbers.
character*3  fnum     Dump/restart file suffix.
character*80 ratel(mchem)
                      String describing volume rates.
character*80 ratesl(mchem)
                      String describing surface rates.
character*80 specl(mspec)
                      String describing  species.

integer anrate        Equal to 8.  Atomic neutralization rate type value
                         for ratet.
integer darate        Equal to 4.  Attachment rate type value for ratet.
integer derate        Equal to 7.  Detachment rate type value for ratet.
integer diarate       Equal to 6.  Dissassociation with negative and
                         positive ion creation rate type value for ratet.
integer dirate        Equal to  5.  Dissassociation rate type value
                         for value ratet.
integer diirate       Equal to 10.  Dissiassociation with ionization rate
                         type value for ratet.
integer erate         Equal to  2.  Electron elastic scattering rate
                         type value for ratet.
integer ic(mrepd,mchem)
                      Index array for labeling each volume reactant and
                         product with a particular species.
integer ics(mrepd,mchem)
                      Index array for labeling each surface reactant and
                         product with a particular species.
integer ifail         Convergence status flag in poisson solver.
integer ifeed         Sets which end of the coil the rf current is input.
```

```
                        If ifeed = 1 voltage is applied to first coil,
                        else if ifeed = 2 voltage is applied to last
                        coil.
integer ifile Counter for current data dump number.
integer ijdenmax(2,mpec)
                     Location of maximum species density.
integer ijfrch_h(2,mspec)
                     Locations of largest fractional change in densities
                        due to a continuity equation.
integer ijfrch_i(2,mspec)
                     Locations of largest fractional change in densities
                        due to chemistry.
integer ijtempmax(2,mpec)
                     Location of maximum electron and ion temperatures.
integer ionn(mspec)  Gives ion species to which a neutral is ionized to.
integer iorder       Determines spatial differencing scheme order for the
                        electron continuity equation.
                        iorder = 1 for upwind differencing.
                        iorder = 2 for second order differencing.
integer irate        Equal to 1.  Ionization rate type value for ratet.
integer irfstepmax   Counter for timesteps per ac bias cycle.
integer isheath      Number of sheath points in analytic sheath model
                        in the EM solver.  (not used at present)
integer istart       If > 0, restart from istart data dump.
                        Otherwise start new run.
integer istep        Time step counter.
integer istepp       Number of time steps since last data output.
integer isteppr      Number of time steps since last call to IHEAT.
integer itcoun       Poisson solver iteration counter.
integer itmaxsip     Maximum number of SIP scheme poisson iterations.
integer itmaxsor     Maximum number of SOR scheme poisson iterations.
integer itused       Poisson solver iteration counter.
integer iunitd       File unit number.
integer jrf          Approximate axial central position of plasma for
                        substrate rf biasing diagnostics.
integer knum         Number of values in rate constant lookup table.
integer mask(mr,mz)  Mask labeling array.  Zero for cells containing
                        plasma. Otherwise its value gives the structure
                        number.
integer maskd(mbound,3)
                     Cell locations and mask numbers for all nmaskd cells.
integer maskl(mbound,3)
```

```
                      Cell locations and mask numbers for all nmaskl cells.
integer maskr(mbound,3)
                      Cell locations and mask numbers for all nmaskr cells.
integer masktype(-1:mmask)
                      Gives type of material for each mask.  Values are:
                         -1 for vacuum region.
                          0 for plasma region.
                          1 for metal (copper).
                        >=2 for dielectrics.
integer masku(mbound,3)
                      Cell locations and mask numbers for all nmasku cells.
integer msteppr       Maximum number of time steps between calls to IHEAT
                         routine.
integer mcycle        Maximum number of substrate rf biasing cycles.
integer mcyclep       Maximum number of substrate rf biasing cycles
                         between data dumps.
integer mnrate        Equal to 9.  Molecular neutralization type value
                         for ratet.
integer mrf           Maximum number of time steps per substrate rf
                         biasing cycles.
integer msip          SIP solver is used every msip time steps, SOR solver
                         for the other time steps.  The first call to the
                         poisson solver routine NEWPOT is done using the
                         SIP scheme.
integer mstep         Maximum number of time steps.
integer mstepp        Maximum number of time steps between data output.
integer nchem         Number of chemical reactions.
integer nchems        Number of surface chemical reactions.
integer ncoil         Number of induction coils.
integer ncycle        Substrate rf biasing cycle counter.
integer ncyclep       Number of substrate rf biasing cycles since last
                         dataldump.
integer negspec       Number of negative ion species.
integer neutspec      Number of neutral species.
integer nmask         Number of structures (masks).
integer nmaskcath     Structure number for cathode.
integer nmaskcoil1    Mask number of first coil.
integer nmaskd        Number of cells directly below all structures.
integer nmaskdome     Mask number for curved dome dielectric structure.
integer nmaskgap      Mask number for vacuum gap between cathode and wall.
integer nmaskl        Number of cells directly to the left of all structures.
integer nmaskr        Number of cells directly to the right of all structures.
integer nmasku        Number of cells directly above all structures.
integer nnmapem(nrdim,nzdim,4)
                      Mapping array for interpolating from fluid to em grid.
integer nnmapfld(nrdim,nzdim,4)
                      Mapping array for interpolating from em to fluid grid.
integer npr           Number of r zones containing plasma.
integer nprod(mchem)  Number of products per chemical reaction.
integer nprods(mchem) Number of surface products per chemical reaction.
integer nprp1         npr + 1.
integer npz           Number of z zones containing plasma.
integer npzp1         npz + 1.
integer nr            Total number of r grid cells.
integer nreact(mchem) Number of reactants per chemical reaction.
integer nreacts(mchem)Number of surface reactants per chemical reaction.
integer nrem          Number of r grid (cell boundary) points in em
                         solver.
integer nrfstep       Substrate rf biasing cycle time step counter.
integer nrfstepmax    Number of timesteps per ac bias cycle.
integer nrfstepmod    Save rf biasing data mod nrfstepmod timesteps.
integer nri           Number of r grid cells in rsheath region.
integer nrm1          nr - 1.
integer nrp1          nr + 1.
integer nrp2          nr + 2.
integer nspec         Number of chemical species.
integer nz            Total number of z grid cells.
integer nzem          Number of z grid (cell boundary) points in em solver.
integer nzi           Number of z grid cells in zsheath regions.
integer nzm1          nz - 1.
integer nzp1          nz + 1.
integer nzp2          nz + 2.
integer ratet(mchem)  Volume rate type.
                         Ionization is type 1.                    (irate)
                         Electron elastic scattering is type 2.   (erate)
                         Excitation with radiative decay is type 3.(xrrate)
                         Dissassociative attachment is type 4.    (darate)
                         Dissassociation is type 5.               (dirate)
                         Dissassociation  with ionization and
                            attachment is type 6.                 (diarate)
                         Detachment is type 7.                    (derate)
                         Atomic neutralization is type 8.         (anrate)
                         Molecular neutralization is type 9.      (mnrate)
                         Dissassociative ionization is type 10.   (diirate)
                         Excitation or dexcitation is type 11.     (xrate)
integer ratest(mchem) Surface rate type.
integer spect(mspec)  Species type.  Values are 1 for electron,
                         2 for positive ion, -2 for negative ion, and
                         3 for neutral.
integer xrate         Equal to 11.  Excitation or dexcitation rate type
                           value for ratet.
integer xrrate        Equal to 3.  Excitation with dexcitation rate
                           type value for ratet.


real*8  accel         Neutral timestep acceleration.
real*8  actden(0:mspec)
                      Actual neutral density after accelerated time
                         step (cm-3).
real*8  adifr(mr,mz,mspec)
                      ADI coefficient for diffusion.
real*8  adifz(mr,mz,mspec)
                      ADI coefficient for diffusion.
real*8  ahydr(mr,mz)  ADI coefficient for hydrodynamics.
real*8  ahydz(mr,mz)  ADI coefficient for hydrodynamics.
real*8  aphir(mr,mz)  Coefficient for time advanced density in poisson
                         solve.
real*8  aphiz(mr,mz)  Coefficient for time advanced density in poisson
                         solve.
real*8  aparam        SIP acceleration parameter.
real*8  area          Plasma boundary area (cm+2).
real*8  bdifr(mr,mz,mspec)
                      ADI coefficient for diffusion.
real*8  bdifrl(mr,mz,mspec)
                      ADI coefficient for diffusion.  This term contains
                         terms for the flux for boundary cells to the
                         left of structures.
real*8  bdifrr(mr,mz,mspec)
                      ADI coefficient for diffusion.  This term contains
                         terms for the flux for boundary cells to the
                         right of structures.
real*8  bdifz(mr,mz,mspec)
                      ADI coefficient for diffusion.
real*8  bdifzd(mr,mz,mspec)
                      ADI coefficient for diffusion.  This term
                         contains terms for the flux for boundary cells
                         down from structures.
real*8  bdifzu(mr,mz,mspec)
                      ADI coefficient for diffusion.  This term
                         contains terms for the flux for boundary
                         cells up from structures.
real*8  bhydr(mr,mz)  ADI coefficient for hydrodynamics.
real*8  bhydrl(mr,mz) ADI coefficient for hydrodynamics.  This term
                         contains terms for the flux for boundary cells
                         to the left of structures.
real*8  bhydrr(mr,mz) ADI coefficient for hydrodynamics.  This term
                         contains terms for the flux for boundary cells
                         to the right of structures.
real*8  bhydz(mr,mz)  ADI coefficient for hydrodynamics.
real*8  bhydzd(mr,mz) ADI coefficient for hydrodynamics.  This term
                         contains terms for the flux for boundary cells
                         down from structures.
real*8  bhydzu(mr,mz) ADI coefficient for hydrodynamics.  This term
                         contains terms for the flux for boundary cells
                         up from structures.
real*8  biascapac Capacitance of capacitor in series with substrate bias
                         voltage source (Farads).
real*8  biasfreq      Substrate holder substrate rf biasing frequency
                         (Hertz).
real*8  biasresistance
                      Bias resistor (Ohm).
real*8  bphir(mr,mz)  Coefficient for time advanced density in poisson
                         solve.
real*8  bphiz(mr,mz)  Coefficient for time advanced density in poisson
                         solve.
real*8  brvoltage     Bias resistor voltage (V).
real*8  cdifr(mr,mz,mspec)
                      ADI coefficient for diffusion.
real*8  brvoltdiag(0:mrfstep)
                      Time history for bias resistor voltage (V).
real*8  capcharge     Capacitor charge (Coul).
real*8  capvoltage    Capacitor voltage (V).
real*8  capvoltdiag(0:mrfstep)
                      Time history for capacitor voltage (V).
real*8  cdifz(mr,mz,mspec)
                      ADI coefficient for diffusion.
real*8  chydr(mr,mz)  ADI coefficient for hydrodynamics.
real*8  chydz(mr,mz)  ADI coefficient for hydrodynamics.
```

```
real*8  coilfreq      RF inductive frequency (Hertz).
real*8  collloss      Volume integrated energy loss due to collisions in
                         electron energy equation (W).
real*8  conchn        Convergence parameter used in NEWPOT, D03EBF (eV).
real*8  conduct(mr,mz)
                      Conductive part of the electron energy flux (W/cm+2/s).
real*8  conductcdiag(0:mrfstep)
                      Time history for substrate conduction current (A).
real*8  conres        SIP residual error limit.
real*8  convect(mr,mz)
                      Convective part of the electron energy flux (W/cm+2/s).
real*8  cour          Courant condition limit factor.  If abs(acbias) = zero,
                         then cour usually determines the time step after
                         the code has gradually increased the time step
                         from the initial value dt0.  A value of cour = 1
                         is conservative.  Values of cour from 5--10
                         should work well.  The electron and ion densities
                         will get noisy for too large values of cour.
real*8  cphir(mr,mz)  Coefficient for time advanced density in poisson
                         solve.
real*8  cphiz(mr,mz)  Coefficient for time advanced density in poisson
                         solve.
real*8  cs(0:mrp1,0:mzp1,mspec)
                      Thermal speed for each species (cm/s).
real*8  csectin(mspec,mspec)
                      Ion-neutral collision cross sections.  First index
                         gives ion, second gives neutral. (cm+2)
real*8  currmask(mmask,mspec)
                      Rate at which each particle species flows to (or from)
                         each masked surface (s-1).
real*8  currmaskall(0:mspec)
                      Rate at which each particle species flows out of (or
                         into) the reactor (summed over all surfaces) (s-1).
real*8  dcbias        DC voltage to substrate holder (V).
real*8  denchem(mspec,mchem,2)
                      Integrated density changes from volume and surface
                         chemistry (#).
real*8  denmax(mspec) Peak density for each species (cm-3).
real*8  den0          Initial peak ion and electron density (cm-3).
real*8  dens(0:mrp1,0:mzp1,mspec)
                      Density of each species (cm-3).
real*8  dielec(mr,mz) Value for relative permittivity.
```

```
real*8  divqc(mr,mz)  Conductive flux divergence term in electron energy
                         equation (W/cm+3).
real*8  divqf(mr,mz)  Convective flux divergence term in electron energy
                         equation (W/cm+3).
real*8  dneutden(mspec)
                      Change in neutral density (cm-3).
real*8  drb(mrp1)     Distances between adjacent radial cell centers (cm).
real*8  drc(mr)       Distances between adjacent radial cell boundaries (cm).
real*8  dt0           Initial time step (s).
real*8  dt            Time step (s).
real*8  dtc           Courant condition time step (s).
real*8  dte           dtefactor * previous time step (s).
real*8  dtefactor     Extrapolation factor for new timestep.  The new time
                         step may be no larger than dtefactor larger than
                         the old time step.  A value of dtefactor close to
                         unity allows a smooth start up from the initial
                         time step dt0.  Not used if abs(acbias) > 0.
real*8  dtifactor     Chemistry time step factor.  Gives approximate
                         maximum fractional change in density per time
                         step due to chemistry.  Not used if
                         abs(acbias) > 0.
real*8  dtchem        Chemistry time step (s).
real*8  dtold         Old time step value (s).
real*8  dtrf          Substrate rf biasing cycle time step.  Equal to
                         the plasma time step if abs(acbias) > 0.  (s)
real*8  dzb(mzp1)     Distances between adjacent axial cell centers (cm).
real*8  dzc(mz)       Distances between adjacent axial cell boundaries (cm).
real*8  echar         Electronic charge (statcoul).
real*8  echarmks      Electronic charge in MKS units (Coulomb).
real*8  echar300      Electronic charge/300 (statcoul).
real*8  echarx300     Electronic charge*300 (statcoul).
real*8  eloss(mchem)  Energy loss per collision for each type of electron
                         collision (eV).
real*8  elosstot(mr,mz)
                      Electron-neutral collisional energy loss term in the
                         electron energy equation (W/cm+3).
real*8  eps(0:mrp1,0:mzp1)
                      Relative permittivity at cell centers.
real*8  epshr(mrp1,mzp1)
                      Relative permittivity on r boundaries.
real*8  epshz(mrp1,mzp1)
                      Relative permittivity on z boundaries.
```

```
real*8  epsilon0      Vacuum permittivity=0.0797 (dimensionless).
real*8  er(mrp1,mzp1) Electric field in r direction on cell boundary
                         (V/cm).
real*8  erold(mrp1,mzp1)
                      Old electric field in r direction on cell boundary
                         (V/cm).
real*8  ez(mrp1,mzp1) Electric field in z direction on cell boundary
                         (V/cm).
real*8  ezold(mrp1,mzp1)
                      Old electric field in z direction on cell boundary
                         (V/cm).
real*8  fhydro(0:mrp1,0:mzp1)
                      Temporary array used in fluid routines.
real*8  flmaskd(mbound)
                      Boundary flux for points corresponding to maskd
                         (cm-2 s-1).
real*8  flmaskl(mbound)
                      Boundary flux for points corresponding to maskl
                         (cm-2 s-1).
real*8  flmaskr(mbound)
                      Boundary flux for points corresponding to maskr
                         (cm-2 s-1).
real*8  flmasku(mbound)
                      Boundary flux for points corresponding to masku
                         (cm-2 s-1).
real*8  flow(mspec)   Neutral feed gas flow rate.  (sccm)
real*8  fluxcr(mrp1,mzp1,mspec)
                      Radial flux at cell center (cm-2 s-1).
real*8  fluxcz(mrp1,mzp1,mspec)
                      Axial flux at cell center (cm-2 s-1).
real*8  fluxdr(mrp1,mzp1,mspec)
                      Radial diffusive flux through cell boundary (cm-2 s-1).
real*8  fluxdz(mrp1,mzp1,mspec)
                      Axial diffusive flux through cell boundary (cm-2 s-1).
real*8  fluxmr(mrp1,mzp1,mspec)
                      Radial drift flux through cell boundary (cm-2 s-1).
real*8  fluxmz(mrp1,mzp1,mspec)
                      Axial drift flux through cell boundary (cm-2 s-1).
real*8  fluxr(mrp1,mzp1,mspec)
                      Radial flux through cell boundary (cm-2 s-1).
real*8  fluxz(mrp1,mzp1,mspec)
                      Axial flux through cell boundary (cm-2 s-1).
```

```
real*8  fmask(mr,mz)  SOR poisson structure characterization array.  If
                         masktype = 1 for the cell, the fmask = 0,
                         otherwise fmask = 1.
real*8  fourpi        4 * pi.
real*8  fracinit(mspec)
                      Initial fractional loading of ions or neutrals.
real*8  fracneut      Fractional change in total neutral density to
                         rescale it back to its original value.
real*8  frch_h(mspec) Largest fractional change in densities from
                         continuity equation.
real*8  frch_hn(mspec)Densities for each species at location of largest
                         fractional change due to continuity equation
                         (cm-3).
real*8  frch_i(mspec) Largest fractional change in densities from
                         chemistry.
real*8  frch_in(mspec)Densities for each species at location of largest
                         fractional change due to chemistry (cm-3).
real*8  fuzz          Very small number = 1.e-20.
real*8  fuzzy         Very, very small number = 1.d-50.
real*8  gammac(mspec) Wall chemistry reaction coefficient.
real*8  gammai(mspec) Wall recombination coefficient for ion corresponding
                         to a given neutral species.
real*8  gammar(mspec) Neutral wall recombination coefficient.
real*8  ionvol(mspec) Volume integrated ionization for each species (s-1).
real*8  ionvoltot(mspec)
                      Time integration of ionvol for each species (#).
real*8  kboltz        Boltzmann constant (eV/K).
real*8  kboltzq       kboltz/echar300 (eV/K/statcoul).
real*8  kinegy        Volume integrated electron thermal energy (erg).
real*8  kinegyf       Volume integrated electron thermal energy at the
                         end of the substrate rf cycle. (erg).
real*8  kinegyi       Volume integrated electron thermal energy at the
                         beginning of the substrate rf cycle. (erg).
real*8  kslope(0:mvalue+1,mchem)
                      Rate constant lookup table slopes (cm+3/s/eV).
real*8  ktemp(0:mvalue+1)
                      Rate constant lookup table temperatures (eV).
real*8  kvalue(0:mvalue+1,mchem)
                      Rate constant lookup table values (cm+3/s).
real*8  me            Electron mass (g).
real*8  minusr(mrp1,mzp1)
                      Negative upwind velocity factor.
```

```
real*8  minusz(mrp1,mzp1)
                      Negative upwind velocity factor
real*8  mob(0:mrp1,0:mzp1,mspec)
                      Mobility of each species (statcoul*s/g).
real*8  mp            Proton mass (g).
real*8  mt            Maximum run time (s).
real*8  mtp           Maximum time between data output (s).
real*8  mtpr          Maximum time between calls to IHEAT subroutine (s).
real*8  newden(0:mrp1,0:mzp1,0:mspec)
                      Temporary density array of each species (cm-3).
real*8  newwden(0:mrp1,0:mzp1,mspec)
                      Temporary n*k*T for each species (eV/cm+3).
real*8  nu(0:mrp1,0:mzp1,mspec)
                      Total electron-neutral collision frequency (s-1).
real*8  ohmic(mr,mz) Joule heating/cooling term in electron energy
                         equation (W/cm+3).
real*8  ohmicloss     Volume integrated Joule heating/cooling term in
                         electron energy equation (W).
real*8  oldden(0:mrp1,0:mzp1,0:mspec)
                      Temporary old density array of each species (cm-3).
real*8  oldphimask(mmask)
                      Old value of structure potential (eV).
real*8  oriden(0:mspec)
                      Original neutral density (cm-3).
real*8  pabs          Total inductive power absorbed in the plasma.  Equals
                         prf if prf > 0 or if(uniform .eqv. .true.) (W).
real*8  pabse         Electron sheath heating from em routines (W).
                         (Not used at present.)
real*8  pabsi         Ion sheath heating from em routines (W).
                         (Not used at present.)
real*8  particlesf(mspec)
                      Total number of particles for each species at the
                         end of the substrate rf cycle (#).
real*8  particlesi(mspec)
                      Total number of particles for each species at the
                         beginning of the substrate rf cycle (#).
real*8  peravgcurr(mspec)
                      Substrate rf period averaged wall current (#/s).
real*8  peravginvol(mspec)
                      Substrate rf period averaged chemistry rate
                         change (#/s).
real*8  peravgohmicloss(mspec)
```

```
                   Substrate rf period averaged ohmicloss (W).
real*8  peravgpowerin(mspec)
                   Substrate rf period averaged coil power (W).
real*8  peravgthermcond(mspec)
                   Substrate rf period averaged wall thermal loss (W).
real*8  phase0     Initial substrate rf biasing phase.
real*8  phi(0:mrp1,0:mzp1)
                   Potential (V).
real*8  phiacc     Fractional change convergence limit for phi for SIP.
real*8  phiacco    Fractional change convergence limit for phi for SOR.
real*8  phimask(nmask)Structure potentials (V).
real*8  phimaskdiag(0:mrfstep)
                   Time history for phimask (V).
real*8  phimax     Maximum of potential values (V).
real*8  phifactd(mbound)
                   Exponential term exp(delta phi/kTe) across sheath
                      for points below of structure surfaces.
real*8  phifactl(mbound)
                   Exponential term exp(delta phi/kTe) across sheath
                      for points to the left of structure surfaces.
real*8  phifactr(mbound)
                   Exponential term exp(delta phi/kTe) across sheath
                      for points to the right of structure surfaces.
real*8  phifactu(mbound)
                   Exponential term exp(delta phi/kT) across sheath
                      for points above structure surfaces.
real*8  phiplasma(0:mrfcount)
                   Plasma potential history at position (1,jrf).
real*8  phir       Reference potential.  Potential added to all applied
                      structures.  Generally phir = 0 used.  (V)
real*8  pi         3.1415926
real*8  plusr(mrp1,mzp1)
                   Positive upwind radial velocity factor.
real*8  plusz(mrp1,mzp1)
                   Positive upwind axial velocity factor.
real*8  powerin    Volume integrated rf power input in electron energy
                      equation (W).
real*8  powerinput(mr,mz)
                   Specified power deposition profile (ergs/cm-3).
real*8  pressure   Total neutral gas pressure (mTorr).
real*8  prf        If(uniform .eqv. .true.) then prf is the inductive
                      power absorbed in the plasma, and the power is
```

```
                         uniformly deposited.  If(uniform .eqv. .false.)
                         then if(prf.ne.0.) prf is the power absorbed
                         non-uniformly and the current iin is scaled to be
                         consistent with prf.  If(prf.eq.0.) iin remains
                         fixed at initial guess and the inductive power
                         is determined by iin.  (W)
real*8  propmask(-1:mask)
                      Defiles relative permittivity for all materials
                         except metals, for which it defiles conductivity.
real*8  qch(0:mrp1,0:mzp1)
                      Space charge density * qeps (statcoul).
real*8  qeps0         echar/epsilon0 (statcoul).
real*8  qm(mspec)     Charge to mass ratio for each species (statcoul/g).
real*8  qmask(mmask,mspec)
                      Time integrated accumulation (or depletion) of each
                         species at each masked surface (#).
real*8  qvol(mspec)   Volume integrated density for each species (#).
real*8  rateconst(0:mrp1,0:mzp1,mchem)
                      Rate constants for each collision type (cm+3/s).
real*8  rateconstgbl(0:mrp1,0:mzp1,mchem)
                      Global rate constants for each collision type (cm+3/s).
real*8  rb(0:mrp2)    Radial cell boundary positions (cm).
real*8  rblock        Radius of blocking dielectric (cm).
real*8  rc(0:mrp1)    Radial cell center positions (cm).
real*8  rcont(mr,mz,mchem)
                      Rate constants times densities (cm-3/s).
real*8  rfavedens(mr,mz,mspec)
                      Period averaged density (cm-3).
real*8  rfavefbr(mrp1,mzp1,mspec)
                      Period averaged radial flux (s-1 cm-2)
real*8  rfavefbz(mrp1,mzp1,mspec)
                      Period averaged axial flux (s-1 cm-2)
real*8  rfavephi(mr,mz)
                      Period averaged potential (V).
real*8  rfavetemp(mr,mz,mspec)
                      Period averaged temperature (eV).
real*8  rfbias(0:mrfstep)
                      RF bias voltage (V).
real*8  rfbiasdiag(0:mrfstep)
                      Bias history for substrate rf biasing voltage (V).
real*8  rfcathcdiag(0:mrfstep)
                      Time history for ac bias rf current (A).
```

```
real*8  rffield(mr,mz,3)
                      Magnitude of azimuthal E field on fluid mesh (V/cm).
real*8  rfphasediag(0:mrfstep)
                      Temporal phase history for substrate rf biasing
                         voltage.
real*8  rmax          Maximum in r extent for fluid routines (cm).
real*8  rmaxem        Total width in radial direction for em routines (m).
real*8  rn(nrdim)     Radial cell boundary positions in em routines (m).
real*8  rplasma       Initial width of plasma in r direction (cm).
real*8  rsheath       Thickness of stretched region in r about rmax
                         radial surface (cm).  If rsheath <= 0 use uniform
                         r grid.  (Use with caution!)
real*8  r1coil(icdim) Inner radial boundary of each inductive coil (m).
real*8  r2coil(icdim) Outer radial boundary of each inductive coil (m).
real*8  sbar(imax)    Electron sheath heating (W/m+2).
                         (Not used at present.)
real*8  scont(mr,mz,0:mspec)
                      Total change due to all chemistry reactions for
                         each species (s-1 cm-3).
real*8  secondary(mspec)
                      Secondary emission coefficients.
real*8  shydro(0:mrp1,0:mzp1)
                      Temporary array used in fluid routines.
real*8  sion(imax)    Ion sheath heating (W/m+2).
real*8  specm(mspec)  Masses of fluid species.  On input units of proton
                         mass used.  specm is then multiplied by the
                         proton mass for internal usage.
real*8  specq(mspec)  Charges of fluid species (units of elementary
                         charge).
real*8  surfchg(mr,mz)Surface charge density on dielectrics (cm-3).
real*8  t             Time (s).
real*8  temp(0:mrp1,0:mzp1,mspec)
                      Temperature of each species (eV).
real*8  tempe         Global electron temperature (eV).
real*8  tempe0        Initial, uniform electron temperature (eV).
real*8  tempi0        Initial, uniform ion temperature (eV).
real*8  tempmax(mspec)Maximum temperature for each species (eV).
real*8  tempn0        Initial, uniform neutral temperature (eV).
real*8  thermcondmask(0:mmask)
                      Thermal energy flux to surfaces (W).
real*4  timei         Initial machine CPU time for timing of simulation
                         run (s).
```

```
real*8  totcollloss   Total accumulated power loss due to collisions
                          during substrate rf cycle (ergs/s).
real*8  totspecnew(mspec)
                         Total # of species in discharge.
                         (For global models.)
real*8  totspecold(mspec)
                          Old total # of species in discharge.
                          (For global models.)
real*8  totohmicloss  Total accumulated power loss due to ohmic heating
                          during substrate rf cycle (ergs/s).
real*8  totpowerin    Total accumulated volume power input during rf cycle
                          (ergs/s).
real*8  totrfcurr     Total accumulated species wall current during rf
                          cycle (#/s)
real*8  totrfinvol    Total accumulated species volume chemistry rate
                          during rf cycle (#/s)
real*8  totthermcond  Total accumulated power loss due to thermal
                          conduction to walls during rf cycle (ergs/s).
real*8  tottrf        Total time since ac bias turned on (s).
real*8  tp            Time since last call to OUTPUT routine (s).
real*8  tpr           Time since last called IHEAT routine (s).
real*8  tprold        Time at which IHEAT was last called (s).
real*8  trf           Time since beginning of ac bias cycle (s).
real*8  vcr(mrp1,mzp1,mspec)
                      Total velocity in r direction at cell center (cm/s).
real*8  vcz(mrp1,mzp1,mspec)
                      Total velocity in z direction at cell center (cm/s).
real*8  vdr(mrp1,mzp1,mspec)
                      Diffusion velocity in r direction at cell boundary.
                         Evaluated for electrons only.  (cm/s)
real*8  vdz(mrp1,mzp1,mspec)
                      Diffusion velocity in z direction at cell boundary.
                         Evaluated for electrons only.  (cm/s)
real*8  vmr(mrp1,mzp1,mspec)
                      Drift velocity in r direction at cell boundary.
                         Evaluated for electrons only.  (cm/s)
real*8  vmz(mrp1,mzp1,mspec)
                      Drift velocity in z direction at cell boundary.
                         Evaluated for electrons only.  (cm/s)
real*8  volume Plasma volume (cm+3).
real*8  vr(mrp1,mzp1,mspec)
                      Total velocity in r direction at cell boundary (cm/s).
```

```
real*8  vz(mrp1,mzp1,mspec)
                      Total velocity in z direction at cell boundary (cm/s).
real*8  v0(imax)      DC potential across sheath from em routines (V).
                         (Not used at present.)
real*8  v1(imax)      RF potential across sheath from em routines (V).
                         (Not used at present.)
real*8  wallchem(mspec)
                      Change in total number of neutrals due to wall
                         chemistry. (#)
real*8  wden(0:mrp1,0:mzp1,mspec)
                      n*k*T for each species (eV/cm+3).
real*8  wdot(nrdim,nzdim)
                      Deposited power/volume on fluid mesh (ergs/s/cm+3).
real*8  wdotfld(0:mrp1,0:mzp1)
                      Deposited power/volume on em routines mesh (W/m+3).
real*8  xkte(nrdim,nzdim)
                      Electron temperature used in em routines. (Joul)
real*8  xmi1          Ion mass used in em routines (kg).
real*8  xne(nrdim,nzdim)
                      Electron density on em routines mesh (m-3).
real*8  xplot(imax)   Distance along the sheath from em routines (m).
                         (Not used at present.)
real*8  wallloss(mspec)
                      Surface chemistry wall loss term (s-1).
real*8  zb(0:mzp2)    Axial cell boundary positions (cm).
real*8  zblock        Axial height of blocking dielectric (cm).
real*8  zc(0:mzp1)    Axial cell center positions (cm).
real*8  zeta          Determines if electron continuity equation solved
                         implicitly or not.  If zeta=1 implicit scheme
                         used, if zeta=0 explicit scheme used.
real*8  zm(nzdim)     Axial  cell boundary positions in em routines (m).
real*8  zmax          Total height in z direction for fluid routines (cm).
real*8  zmaxem        Total height in axial direction for em routines (m).
real*8  zplasma       Initial height of plasma in z direction (cm).
real*8  zsheath       Thickness of stretched region in z about 0 and zmax
                      surfaces (cm).  If zsheath <= 0 use uniform z grid.
real*8  z1coil(icdim) Lower axial boundary of each inductive coil (m).
real*8  z2coil(icdim) Upper axial boundary of each inductive coil (m).


complex*16  acbias    Phasor rf voltage to substrate holder (V).
complex*16  ercap(nrdim,nzdim)
```

```
                      Phasor radial electric field from em routines (V/m).
complex*16  eth(nrdim,nzdim)
                      Phasor azimuthal electric field from em routines (V/m).
complex*16  ezcap(nrdim,nzdim)
                      Phasor axial electric field from em routines (V/m).
complex*16  icurr(icdim)
                      Phasor stray capacitive current from each coil (A).
complex*16  iin       Phasor rf input current to inductive coil (A).
complex*16  phicap(nrdim,nzdim)
                      Phasor electric potential from em routines (V).
complex*16  phicapfld(0:mrp1,0:mzp1)
                      Phasor rf coil potential on fluid grid (V).
complex*16  sigmap(nrdim,nzdim)
                      Plasma conductivity on em routines mesh (mho/m).
complex*16  sigmapfld(0:mrp1,0:mzp1)
                      Plasma conductivity on fluid mesh (1/s).
complex*16  vcoil(icdim)
                      Inductive coil voltages calculated by em routines (V).
complex*16  vcoilold(icdim)
                      Inductive coil voltages at time of last call to
                         iheat (V).
complex*16  vl(icdim) Phasor inductive voltage drop across each coil (V).
complex*16  vout      Phasor output voltage at coil termination (V).
complex*16  xicoil(icdim)
                      Inductive coil currents calculated by em routines (A).
complex*16  zout      Impedance from last coil to ground (ohms).
```

2.1.6 SID94.INC

This file contains common blocks and variable definitions used by the SIP poisson solver. The workspace variables are used as scratch arrays elsewhere in the code. Variable definitions are reproduced here.

```
 Common Block Variable Descriptions
***************************************************************************
 real*8  a            Poisson equation coupling coefficient to cell i,j-1.
 real*8  b            Poisson equation coupling coefficient to cell i-1,j.
 real*8  c            Poisson equation coupling coefficient to cell i,j.
 real*8  d            Poisson equation coupling coefficient to cell i,j+1.
 real*8  e            Poisson equation coupling coefficient to cell i+1,j.
 real*8  chngs        Iteration array of maximum change in potential
                         during SIP solve (V).
 real*8  resids       Iteration array of maximum Poisson residual
```

```
                           from SIP solve.
 real*8  res         Matrix of residuals for SIP solve.
 real*8  wrksp1      SIP workspace array.
 real*8  wrksp2      SIP workspace array.
 real*8  wrksp3      SIP workspace array.
 real*8  wrksp4      SIP workspace array.
 real*8  wrksp5      SIP workspace array.
```

2.2 Subroutines

INDUCT93 was written in FORTRAN using FORTRAN77, with selected extensions. It was designed to run on a variety of computers with few machine specific constructs. The only specific machine dependent call is to the IBM RS/6000 timing routine MCLOCK which is called from the subroutine TIMER. No graphics routines have been incorporated into INDUCT94. All graphics is currently post-processed from files written for the dual purpose of graphical analysis and of restarting the simulation.

The fluid portion of INDUCT94 code uses cgs-gaussian units. The coil rf field solver (EMSOURCE subroutine and subroutines called by EMSOURCE) uses MKS units. Most input parameters used by the rf field solver are input in cgs, and are then converted to MKS in the SETUP subroutine. See the input variable descriptions following the namelist statement in the SETUP subroutine.

We list each subroutine and function here alphabetically, with a very brief description of its use. Figure Figure 4

2.2.1 subroutine ABSORB – *EM Routine*

This routine Calculates the local energy deposition WDOT(r,z) and integrates over the r-z plane to get power absorbed PABS. Also calculates the local power transferred from the antenna and integrates over the r-z plane to get power transferred from the antenna PC.

Dummy Variables:

ER: Complex radial electric field (V m^{-1}).

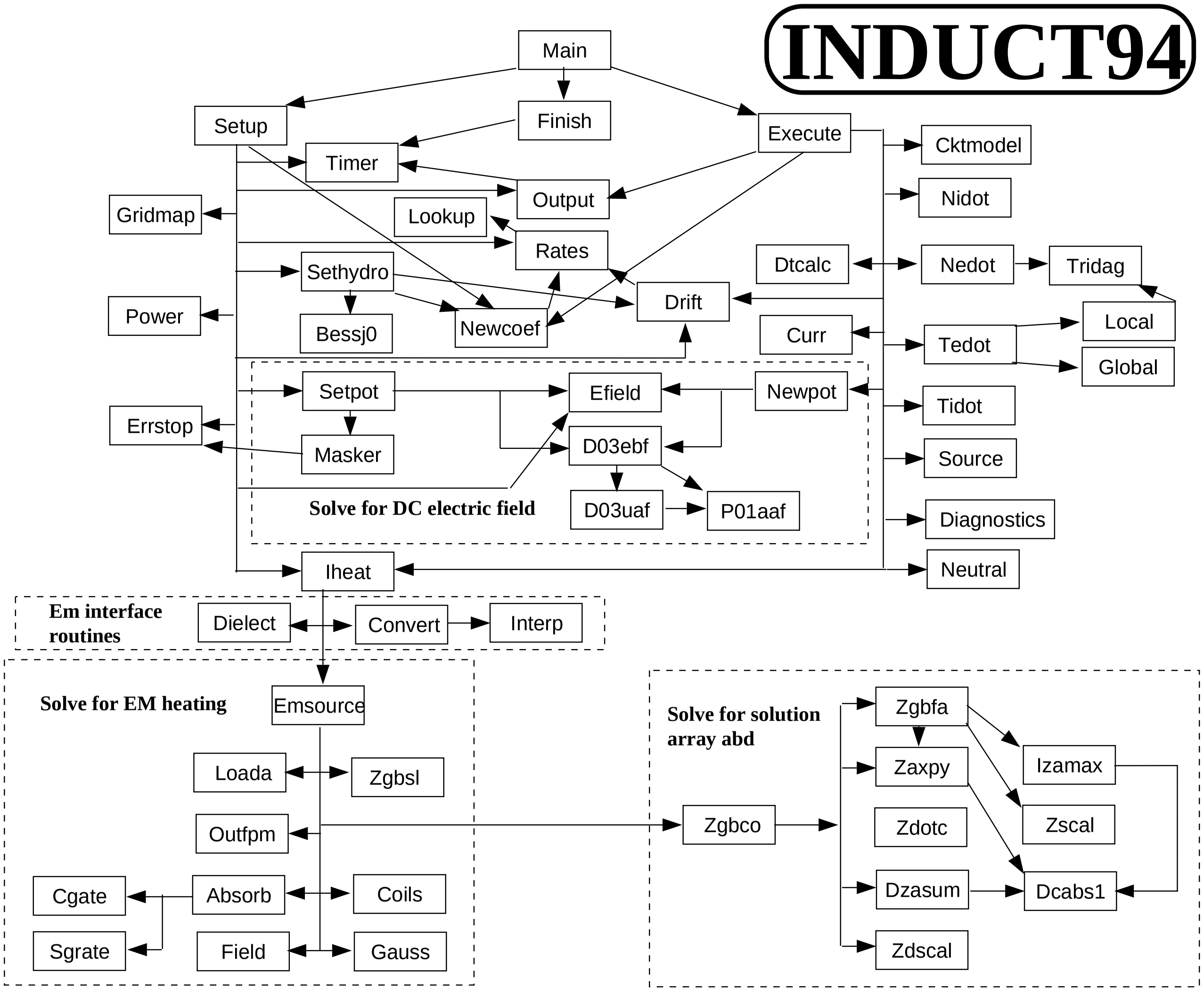
ETH: Complex θ electric field (V m^{-1}).

EZ: Complex axial electric field (V m^{-1}).

XJR: Complex radial plasma current density (m^{-3}).

XJTH: Complex θ plasma current density.

XJZ: Complex axial plasma current density.


INDUCT94
Main
Finish
Execute
Setup
Timer
Output
Gridmap
Lookup
Rates
Sethydro
Bessj0
Newcoef
Drift
Power
Dtcalc
Curr
Cktmodel
Nidot
Nedot
Tridag
Tedot
Local
Global
Tidot
Source
Diagnostics
Neutral
Setpot
Masker
Errstop
Efield
D03ebf
Newpot
D03uaf
P01aaf
Solve for DC electric field
Iheat
Em interface routines
Dielect
Convert
Interp
Solve for EM heating
Emsource
Loada
Zgbsl
Outfpm
Cgate
Absorb
Coils
Sgrate
Field
Gauss
Solve for solution array abd
Zgbco
Zgbfa
Zaxpy
Zdotc
Dzasum
Zdscal
Izamax
Zscal
Dcabs1

XKPRP: Complex plasma dielectric.

NR: Number of radial mesh points.

NZ: Number of axial mesh points.

RN: Radial position (m).

XALPHA: Not used.

WDOT: Real power deposition at (ir,iz) (W m^{-3}),

WDOT2: Local energy transferred (W m^{-3}).

PABS: Total power absorbed by the plasma (W).

PC: Total power transferred from the antenna (W).

ZM: Axial position (m).

RWDOT: Not used.

RWDT2R: Not used.

RWDT2I: Not used.

BZ: Axial magnetic field.

BR: Radial magnetic field.

NDOPED: Doped silicon index matrix.

WDOTDS: Power absorbed by electron in sheath. Not used.

WDOTPL: Power absorbed by ions in sheath. Not used.

NRMAX: Maximum radial dimension index.

NZMAX: Maximum axial dimension index.

2.2.2 double precision function BESSI0 – *EM Routine*

This function returns the modified Bessel function Io(x) for any real x.

Dummy Variables:

X: Argument of Bessel function.

2.2.3 double precision function BESSJ0 – *EM Routine*

This function returns the Bessel function J0(x) for any real X.

Dummy Variables:

X: Argument of Bessel function.

2.2.4 block data

This is the block data routine for initializing variables in common blocks.

2.2.5 subroutine CGRATE – *EM Routine*

This function does a complex function integration.

Dummy Variables:

F: Function to integrate.

NR: Number of radial grid points.

NZ: Number of axial grid points.

ANS: Result.

R: Radial position.

NRMAX: Maximum radial dimension index.

NZMAX: Maximum axial dimension index.

2.2.6 subroutine CHECKU – *EM Routine*

Error checking routine for EM routines.

Dummy variables:

NR: Number of radial mesh points.

NZ: Number of axial mesh points.

U: Complex potential.

V: Complex ETH/RN.

XKPRP: Complex plasma dielectric.

RN: Radial position.

RNP: Radial position - DR/2.

RNM: Radial position + DR/2.

XALPHA: Not used.

BZ: Axial magnetic field.

BR: Radial magnetic field.

ISWEEP: Check index.

XJR: Complex radial plasma current density.

XJTH: Complex theta plasma current density.

XJZ: Complex axial plasma current density.

RHOEXT: External space charge density.

NRMAX: Maximum radial dimension index.

NZMAX: Maximum axial dimension index.

2.2.7 subroutine CHECKV – *EM Routine*

Error checking routine for EM routines.

Dummy variables:

NR: Number of radial mesh points.

NZ: Number of axial mesh points.

U: Complex potential.

V: Complex ETH/RN.

XKPRP: Complex plasma dielectric.

RN: Radial position.

RNP: Radial position - DR/2.

RNM: Radial position + DR/2.

XALPHA: Not used.

BZ: Axial magnetic field.

BR: Radial magnetic field.

ISWEEP: Check index.

XJR: Complex radial antenna current density.

XJTH: Complex theta antenna current density.

XJZ: Complex axial antenna current density.

RHOEXT: External space charge density.

NRMAX: Maximum radial dimension index.

NZMAX: Maximum axial dimension index.

2.2.8 subroutine CKTMODEL

This routine calculates the substrate bias voltage based upon a circuit model. Called by EXECUTE.

2.2.9 subroutine COILS – *EM Routine*

Sets up coils for iteration.

Dummy Variables:

ICDIM: Dimension parameter for number of coils.

NCOIL: Number of coils.

R: Radius.

Z: Axial position.

DR: Radial grid spacing.

DZ: Axial grid spacing.

IFEED: If IFEED = 1 voltage is applied to first coil. Elseif IFEED = 2 voltage is applied to last coil.

R1COIL(i): Inner radius of coil(i).

R2COIL(i): Outer radius of coil(i).

Z1COIL(i): Lower boundary of coil(i).

Z2COIL(i): Upper boundary of coil(i).

XIANT: Complex current on the ith coil (A).

VCOIL: Coil voltage (V).

XJTH: Current density (A m^2).

POTANT: Complex current at grid point (V).

ICOIL: Coil index.

2.2.10 subroutine CONVERT

This routine converts between the fluid and EM grids and units. Called by IHEAT.

Dummy Variables:

NINTERP: Determines whether conversion is to or from fluid grid and units.

2.2.11 subroutine D03UAF

This routine does an iteration for the poisson solve. Called by D03EBF.

Dummy variables:

N1: Number of points to be solved for in the r direction.

N1M: Dimension of r direction arrays.

N2: Number of points to be solved for in the z direction.

APARAM: Acceleration parameter.

IT: Iteration counter.

IFAIL: Failure flag.

2.2.12 subroutine D03EBF

This routine solves the poisson equation. Called by SETPOT and NEWPOT.

Dummy variables:

N1: Number of points to be solved for in the r direction.

N1M: Dimension of r direction arrays.

N2: Number of points to be solved for in the z direction.

QCH: Source term for Poisson's equation.

T: Potential to be solved for.

APARAM: Acceleration parameter.

ITMAX: Maximum number of SIP scheme Poisson iterations.

ITCOUN: Poisson solver iteration counter.

ITUSED: Poisson solver iteration counter.

NDIR: SIP solver parameter.

IRN: SIP solver parameter.

IZN: SIP solver parameter.

CONRES: SIP residual error limit.

CONCHN: Potential convergence parameter.

IFAIL: Failure flag.

2.2.13 subroutine DIAGNOSTICS

This routine calculates diagnostics and prints global data at end of each rf cycle if rf bias is applied. Called by EXECUTE.

2.2.14 subroutine DIELECT

This routine calculates the complex scaler plasma conductivity in cgs units of 1/sec. Called by IHEAT.

2.2.15 subroutine DRIFT

This routine calculates electron drift velocities. Called by EXECUTE, SETUP, and SETHYDRO.

2.2.16 double precision function DZASUM – *EM Routine*

This function takes the sum of the absolute values.

Dummy variables:

N: Array length.

ZX: Vector.

INCX: Increment.

2.2.17 subroutine DTCALC

This routine updates the time step data, calculates diagnostics, and calculates time dependent variables. Called by EXECUTE.

2.2.18 double precision function DZASUMK – *EM Routine*

This function takes the sum of the absolute values.

Dummy Variables:

N: Sum limit.

ZX: Variable to sum.

INCX: Sum increment.

2.2.19 subroutine EFIELD

This routine calculates the electric field. Called by SETPOT, SETUP, and NEWPOT.

2.2.20 subroutine EMSOURCE – *EM Routine*

This subroutine solves the Helmholtz wave equation and Poisson's equation in conservative form using a direct matrix solution technique in cylindrical geometry. Symmetry in theta is assumed. The subroutine iterates to find the self-consistent coil currents and voltages. All units are MKS.

The program loads the matrix abd for use with linpack routines CBGCO and CGBSL and then solves the matrix equation.

Runs in double precision (64 bit words) on IBM RISC 6000 or single precision on Cray

Dummy variables:

INPUT VARIABLES

NR: Number of radial mesh points

RMAX: Maximum r value in m.

NZ: Number of axial mesh points.

ZMAX: Maximum z value in m.

NCOIL: Number of coils.

R1COIL(i): Inner radii of ith coil.

R2COIL(i): Outer radii of ith coil.

Z1COIL(i): Lower boundaries of ith coil.

Z2COIL(i): Upper boundaries of ith coil.

IFEED: If IFEED =1 voltage is applied to first coil. Elseif IFEED = 2 voltage is applied to last coil.

PRF: Total inductive power absorbed. If PRF .ne. 0, IIN is scaled to give PRF = power absorbed. Elseif PRF = 0, IIN remains fixed at initial guess.

IIN: Initial guess for complex current entering the driven coil (A).

ZOUT: Output impedance terminating the coil.

VBIAS: Complex bias voltage on cathode (V).

FREQ: rf frequency in hertz.

SIGMAP(ir,iz): Complex plasma conductivity at mesh ir,iz (ohm^{-1} m^{-1}).

XNE(ir,iz): Plasma density at mesh ir,iz (m^{-3}).

XKTE(ir,iz): Plasma thermal energy (kT) at mesh ir,iz (Joules).

XMI1: Ion mass (kg).

OUTPUT VARIABLES

VOUT: Output voltage at coil termination.

ERCAP(ir,iz): Complex radial electric field at (ir,iz) (V m^{-1}).

ETH(ir,iz): Complex azimuthal electric field at (ir,iz) (V m^{-1}).

EZCAP(ir,iz): Complex axial electric field at (ir,iz) (V m^{-1}).

PHI(ir,iz): Complex electrostatic potential at (ir,iz) (V).

WDOT(ir,iz): Real power deposition at (ir,iz) (W m^{-3}).

PABS: Total power absorbed by the plasma (W).

PABE: Electron sheath heating (Watts) (not used at present).

PABSE: Ion sheath heating (Watts) (not used at present)..

XIANT: Complex current on the ith coil (A).

VC(i): Complex voltage on the ith coil (V).

VL(i): Inductive voltage drop (complex) across the ith coil (V).

IC(i): Stray capacitive current (complex) from the ith coil (A).

RN(ir): Radius at the (ir,iz) mesh point (m).

ZM((ir,iz)): Axial position at the (ir,iz) mesh point (m).

ISHEATH: Number of points along the sheath (not used at present).

XPLOT(ISHEATH): Distance along the sheath (m) (not used at present)..

V0(ISHEATH): DC potential across the sheath (V) (not used at present)..

V1(ISHEATH): RF potential across the sheath (V) (not used at present)..

SBAR(ISHEATH): Electron sheath heating (W m^{-2}) (not used at present)..

SION(ISHEATH): Ion sheath heating (W m^{-2}) (not used at present)..

2.2.21 subroutine EXECUTE

This routine governs the code execution. Called by MAIN.

2.2.22 subroutine FIELD – *EM Routine*

This routine calculates the capacitive er and ez fields for EN field solver.

Dummy Variables:

PHI: Potential.

ER: Complex radial electric field.

EZ: Complex axial electric field.

RN(ir): Radius at the (ir,iz) mesh point (m).

ZM(ir,iz): Axial position at the (ir,iz) mesh point (m).

NR: Number of radial mesh points.

NZ: Number of axial mesh points.

DR: Radial grid spacing.

DZ: Axial grid spacing.

NRMAX: Maximum radial dimension index.

NZMAX: Maximum axial dimension index.

2.2.23 subroutine FINISH

This routine finishes any calculations at the end. Called by MAIN.

2.2.24 subroutine GAUSS – *EM Routine*

This routine uses Gauss's Law to calculate coil currents.

Dummy variables:

ICDIM: Dimension parameter for number of coils.

NRDIM: Dimension parameter for R.

NZDIM: Dimension parameter for Z.

NR: Number of radial mesh points.

NZ: Number of axial mesh points.

ICOIL: Coils index.

ER: Complex radial electric field.

EZ: Complex axial electric field.

RN: Radius at the (ir,iz) mesh point (m).

DR: Radial grid spacing.

DZ: Axial grid spacing.

XINT: Coil current.

2.2.25 subroutine GLOBAL

This routine calculates the electron temperature from a global electron energy balance equation. Called by EXECUTE.

2.2.26 subroutine GRIDMAP

This routine generates arrays of nearest neighbor gridpoints for interpolating between the fluid and EM meshes. Called by SETUP.

2.2.27 subroutine IHEAT

This routine is the interface program between the fluid routines and the em field solver routines. Called by SETUP and EXECUTE.

2.2.28 program INDUCT94

This is the main routine.

2.2.29 subroutine INTERP

This routine does bilinear interpolation between the fluid and EM grids. Unit conversion between cgs and MKS is done on the fly. Called by CONVERT.

Dummy Variables:

TEMPFLD: Quantity on the fluid grid.

TEMPEM: Quantity on em grid.

NINTERP: Determines whether conversion is to or from fluid grid and units.

2.2.30 integer function IZAMAX – *EM Routine*

This function finds the index of element having maximum absolute value.

Dummy variables:

N: Array length.

ZX: Vector.

INCX: Increment.

2.2.31 subroutine LOADA –*EM Routine*

This routine load finite difference matrix abd.

Dummy Variables:

ABD: Matrix to be loaded.

LDA: First dimension of abd (rows).

NMAT: Second dimension of abd (columns).

NZ: Number of axial grid points.

NR: Number of radial grid points.

E: Plasma dielectric terms.

A: Zero.

ENP: (E(N+1,M)+E(N,M))/2.D0.

ENM: (E(N-1,M)+E(N,M))/2.D0.

EMP: (E(N,M+1)+E(N,M))/2.D0.

EMM: (E(N,M-1)+E(N,M))/2.D0.

RN: Radial position.

RNM: Radial position - DR/2.

RNP: Radial position + DR/2.

XALPHA: Not used.

XJTH: Complex theta antenna current density.

NRMAX: Maximum radial dimension index.

NZMAX: Maximum radial dimension index.

IMAT: Material type matrix.

2.2.32 subroutine LOCAL

This routine calculates the local electron temperature from the electron energy balance equation. Called by EXECUTE.

2.2.33 subroutine LOOKUP

Given a temperature x, returns a value JLO such that X is between KTEMP(JLO) and KTEMP(JLO+1). KTEMP must be monotonic increasing. JLO=0 or JLO=N is returned to indicate that x is out of range. JLO on input is taken as the initial guess for JLO on output. Adapted from subroutine HUNT [12]. Called by rates.

Dummy Variables:

N: Array length.

X: Temperature.

JLO: Array index corresponding to X.

2.2.34 subroutine MASKER

This routine calculates the mask, which specifies boundaries to the plasma and defines structures. Called by SETPOT.

2.2.35 subroutine NEDOT

This routine solves the electron continuity equation using an implicit conservative scheme. Called by EXECUTE.

2.2.36 subroutine NEWCOEF

This routine calculates collision rates and transport coefficients. Called by EXECUTE and SETHYDRO.

2.2.37 subroutine NIDOT

This routine does the hydrodynamics for ions. It solves both the ion continuity equation and the ion momentum equations using explicit time differencing. Called by EXECUTE.

2.2.38 subroutine NEUTRAL

This routine calculates changes in the neutrals. Called by EXECUTE.

2.2.39 subroutine NEWPOT

This routine calculates the non-linear potential using either the SOR or SIP technique. Note that the rf bias boundary conditions are set to the future value of the potential in DTCALC, and the solution obtained from the SOR or SIP solver is the future value of the potential. Called by EXECUTE.

2.2.40 subroutine OUTFPM – *EM Routine*

This routine write out to unit 33 the solutions ETH and PHI.

Dummy Variables:

B:	Source vector for EM solve.
NMAT:	Second dimension of abd (columns).
U:	Complex potential.
V:	Complex ETH/RN.
ETH:	Complex theta electric field.
PHI:	Complex potential.
RN:	Radial position.
NR:	Number of radial grid points.
NZ:	Number of axial grid points.
NRMAX:	Maximum radial dimension index.
NZMAX:	Maximum axial dimension index.

2.2.41 subroutine OUPUT

This routine does data output processing. Data is printed to the output file INDUCT94.out, to the restart file INDUCT94.RST, and to the dumpfiles for restarting and for postprocessing. Called by SETUP and EXECUTE.

2.2.42 integer function P01AAF

Poisson equation error return function.

Dummy Variables:

IFAIL: Failure flag.

IERROR: Error type.

SRNAME: Subroutine name.

2.2.43 subroutine POWER

This routine calculates grid power law position values. Called by SETUP.

Dummy Variables:

DELTA: Initial grid size in power law region.

N: Number of cells in power law region.

ALPHAN: Grid power law factor to be solved for.

2.2.44 subroutine RATES

This routine looks up collision rate constants. Called by SETUP and NEWCOEF.

2.2.45 subroutine SETHYDRO

This routine sets up the initial velocities, fluxes, densities, and temperatures. Called by SETUP.

2.2.46 subroutine SETPOT

This routine sets up the vacuum potential. Structures are setup through the call to masker. Called by SETUP.

2.2.47 subroutine SETUP

This is the primary initialization routine. Input files are read, and echoed to the output file. The grid is set up. Timing is initialized. Parameters are calculated. Variables are initialized through calls to SETPOT, SETHYDRO, and IHEAT, or through reading restart information. Called by MAIN.

2.2.48 subroutine SGRATE – *EM Routine*

This function does a scaler function integration.

Dummy Variables:

F: Function to integrate.

NR: Number of radial grid points.

NZ: Number of axial grid points.

ANS: Result.

R: Radial position.

NRMAX: Maximum radial dimension index.

NZMAX: Maximum axial dimension index.

2.2.49 subroutine TEDOT

This routine calculates electron temperature from the electron energy balance equation. Called by EXECUTE.

2.2.50 subroutine TIDOT

This routine calculates ion temperature from the ion energy balance equation. Called by EXECUTE.

2.2.51 subroutine TRIDAG

This routine solves tri-diagonal systems of equations.

2.2.52 subroutine SOURCE

This routine calculates change in electron and ion density due to chemistry. Called by EXECUTE.

2.2.53 subroutine ZAXPY – *EM Routine*

Calculates constant times a vector plus a vector.

Dummy variables:

N: Array length.

ZA: Constant.

ZX: Vector.

INCX: Increment for ZX.

ZY: Vector.

INCY: Increment for Zy.

2.2.54 double complex function ZDOTC – *EM Routine*

This function forms the dot product of a vector.

Dummy variables:

N: Array length.

ZX: Vector.

INCX: Increment for ZX.

ZY: Vector.

INCY: Increment for ZY.

2.2.55 subroutine ZGBCO – *EM Routine*

This subroutine factors a complex*16 band matrix by gaussian elimination and estimates the condition of the matrix.

Dummy Variables:

INPUT VARIABLES

ABD: Contains the matrix in band storage. the columns of the matrix are stored in the columns of abd and the diagonals of the matrix are stored in rows ML+1 through 2*ML+MU+1 of ABD.

LDA: The leading dimension of the array ABD.

N: The order of the original matrix.

ML: Number of diagonals below the main diagonal.

ML: Number of diagonals above the main diagonal.

OUTPUT VARIABLES

ABD: An upper triangular matrix in band storage and the multipliers which were used to obtain it. the factorization can be written a = l*u where l is a product of permutation and unit lower triangular matrices and u is upper triangular.

IPVT: An integer vector of pivot indices.

RCOND: An estimate of the reciprocal condition of a. for the system a*x = b , relative perturbations in a and b of size epsilon may cause relative perturbations in x of size epsilon/rcond .

Z: A work vector whose contents are usually unimportant.

2.2.56 subroutine ZGBFA – *EM Routine*

This routine factors a complex*16 band matrix by elimination.

Dummy variables:

INPUT VARIABLES

ABD: Contains the matrix in band storage. The columns of the matrix are stored in the columns of ABD and the diagonals of the matrix are stored in rows ML+1 through 2*ML+MU+1 of ABD.

LDA: The leading dimension of the array ABD.

N: The order of the original matrix.

ML: Number of diagonals below the main diagonal.

MU: Number of diagonals above the main diagonal.

OUTPUT VARIABLES

ABD: An upper triangular matrix in band storage and the multipliers which were used to obtain it. The factorization can be written a = l*u where l is a product of permutation and unit lower triangular matrices and u is upper triangular.

IPVT: An integer vector of pivot indices.

INFO: Equals 0 for normal value, equal k if u(k,k) .eq. 0.0 . This is not an error condition for this subroutine, but it does indicate that zgbsl will divide by zero if called. Use rcond in zgbco for a reliable indication of singularity.

2.2.57 subroutine ZGBSL – ***EM Routine***

This routine solves the complex*16 band system a * x = b or ctrans(a) * x = b using the factors computed by zgbco or zgbfa.

Dummy variables:

INPUT VARIABLES

ABD: The output from ZGBCO or ZGBFA.

LDA: The leading dimension of the array ABD.

N: The order of the original matrix.

ML: Number of diagonals below the main diagonal.

MU: Number of diagonals above the main diagonal.

IPVT: The pivot vector from ZGBCO or ZGBFA.

B: The right hand side vector.

JOB: Equals 0 to solve a*x = b, equals nonzero to solve ctrans(a)*x = b, where ctrans(a) is the conjugate transpose.

OUTPUT VARIABLES

B: The solution vector x.

2.2.58 subroutine ZSCAL – *EM Routine*

This routine scales a vector by a constant.

Dummy variables:

N: Array length.

ZA: Constant.

ZX: Vector.

INCX: Increment

2.3 INDUCT94 Output

INDUCT94 produces numerous output files used both for diagnostic purposes and for restarting the model. Text output consists of information printed to the terminal, the file INDUCT94.OUT, and the file GLOBAL.OUT. The file INDUCT94.OUT is the primary output file and is discussed in detail below in the section describing a sample test problem. For simulations with finite ac biasing to the substrate holder, the file GLOBAL.OUT give for each ac bias period and for each species: the ac bias period averaged particle flux to all boundary surfaces for each species; the ac bias period averaged volume integrated ionization rate; and the change in total number of particles over the period.

The FORTRAN binary output from a INDUCT94 simulation consists of the restart file INDUCT94.RST, the ac bias data file INDUCT94.RF, and the dumpfile data files DEN##, PHI##, TEM##, POW##, FBR##, FBZ##, FCR##, FCZ##, VBR##, VBZ##, VCR##, VCZ##, FLD##, SIG##, RIZ##, and SCH##. The dumpfiles correspond respectively to the density at grid centers [cm^{-3}], potential at grid centers [eV], temperature at grid centers [eV], grid boundary flux in the z direction [cm^{-2}], grid boundary flux in the r direction [cm^{-2}], grid center flux in the r direction [cm^{-2}], grid center flux in the z direction [cm^{-2}], grid boundary velocity in the r direction [cm s^{-1}], grid boundary velocity in the z direction [cm s^{-1}], grid center velocity in the r direction [cm s^{-1}], grid center velocity in the z direction [cm s^{-1}], the rf heating power density at the grid center [erg cm^{-3}], the rf fields at the grid centers [eV cm^{-1}], the conductivity at the grid centers [s^{-1}], the ionization rate at the grid centers [cm^{-3} s^{-1}], and the surface space charge on dielectric surfaces [cm^{-3}]. For electrons the flux at the cell boundary is the basic flux and the cell center flux is an averaged. For ions, the cell center flux is solved for and the cell boundary flux is the derived value. Velocities are calculated as diagnostic quantities and may show not unphysical behavior at the boundaries. We use ## here to designate the data dump number. If MULTIDUMP is false only one file for each quantity is written, and it is over written each time the subroutine OUTPUT is called. These files have no suffix with dump number. The data files are used both for restarting the code and for graphical analysis of the simulation. The format of the DEN## data files is shown by the following FORTRAN lines

```
      open(10,file=’DEN##’,form=;’unformatted’)
      write(10) (((DENS(i,j,n),i=1,NPR),j=1,NPZ),n=1,NSPEC),
     .   (RC(i),i=1,NR),(RB(i),i=1,NPRP1),
     .   (ZC(j),j=1,NZ),(ZB(j),j=1,NPZP1)
      close(10)}
```

The SCH##, TEM##, FCR##, FCZ##, VCR##, and VCZ## file formats are the same as for the DEN## files. The format for the FBR## and FBZ## files are given by

```
      open(10,file=’FBR##’,form=;’unformatted’)
      write(10) (((FLUXR(i,j,n),i=1,npr),j=1,npzp1),n=1,nspec),
     .   (RC(i),i=1,NPR),(RB(i),i=1,NPRP1),
     .   (ZC(j),j=1,NPZ),(ZB(j),j=1,NPZP1)
      close(10)}
```

The PHI## and POW## file formats are given by

```
      open(10,file=’PHI##’,form=;’unformatted’)
      write(10) ((PHI(i,j),i=1,NPR),j=1,NPZ),
     .   (RC(i),i=1,NPR),(RB(i),i=1,NPR),
     .   (ZC(j),j=1,NPZ),(ZB(j),j=1,NPZ)
      close(10)}
```

The FLD## file formats are given by

```
      open(10,file=’fld##’,form=;’unformatted’)
      write(10) (((FLD(i,j,n),i=1,NPR),j=1,NPZ),n=1,3)
     .   (RC(i),i=1,NPR),(RB(i),i=1,NPRP1),
     .   (ZC(j),j=1,NPZ),(ZB(j),j=1,NPZP1)
      close(10)}
```

If LBIAS is set to be .true. on input, then ac bias period averaged binary data files are generated along with the normal data dump files. These period averaged files begin with “rfave”, and have the same file format as the snap-shot data files.

For diagnostic purposes the most useful data files are for density, temperature, potential, power deposition, and fluxes. Ions flow tends to be a fairly straight forward, showing diverging expansion from the region of maximum density. For electrons, if there is a significant temperature gradient, then there may be a strong circulation flow. This flow is divergent free and indicates a swirling of electrons contained within the reactor by the sheath potential. Non-physical values of data variables may be found outside of the plasma chamber and have no physical significance.

We supply the script file WAVE.PLT for post-processing the sample test problem (see below where the sample test problem is discussed). WAVE.PLT contains plotting commands in the IDL (or PV-WAVE) graphic language. For grid dimensions which differ from the sample problem, WAVE.PLT must be modified.

The ac bias binary data file, INDUCT94.RF, gives time dependent data primarily concerning the external circuit. Variables written to this file include: RFPHASE, RFBIAS, PHIMSASKDIAG, PHIPLASMA, CAPVOLTDIAG, BRVOLTDIAG, RFCATHCDIAG, and CONDUCTCDIAG. The format used to write the file is:

```
      write(19) nrfstepdiag,(rfphase(i),i=0,nrfsetpdiag),
     .                      (rfbias(i),i=0,nrfstepdiag),
     .                      (phimaskdiag(i),i=0,nrfstepdiag)
     .                      (phiplasma(i),i=0,nrfstepdiag)
     .                      (capvoltdiag(i),i=0,nrfstepdiag)
     .                      (brvoltdiag(i),i=0,nrfstepdiag)
     .                      (rfcathcdiag(i),i=0,nrfstepdiag)
     .                      (conductcdiag(i),i=0,nrfstepdiag)
```

Note

3 SIMULATION INITIALIZATION

The model initialization, or setup, for INDUCT94 takes place as the first order of business. Initialization is handled through the subroutine SETUP, which is called by the main routine. The fluid portion of INDUCT94 uses cgs-gaussian units. The EM solver (EMSOURCE subroutine) uses MKS units. Most input parameters used by the EM solver are input in cgs, and are then converted to MKS in the SETUP subroutine. SETUP in turn calls the following routines:

TIMER: This routine does the problem timing. *WARNING!!!!!! WARNING!!!!!!* The timing routine calls the system routine MCLOCK which is an IBM RS/6000 system routine.

GRIDMAP: This routine generates arrays of nearest neighbor gridpoints for interpolating between the fluid and EM meshes. Interpolation is done using a bi-linear interpolation scheme.

POWER: This routine calculates grid power law values if a stretched mesh is desired.

OUTPUT: This routine does data output to both the restart file, and to the dumpfiles for restarting and for postprocessing.

RATES: This routine calculates collision rate constants for ionization, collisional excitation cooling, and, momentum transfer.

SETHYDRO: This routine sets up the initial velocities, fluxes, densities, and temperatures for the fluid model.

SETPOT: This routine sets up the initial DC potential. A vacuum potential is solved for here as the plasma is taken as neutral at time = 0.

IHEAT: This routine is the interface program between the fluid routines and the EM field solver routines. Currently only the rf power deposition rate calculated by the EM routines is used. The heating rate is typically held constant over many time steps due to the extensive computational time currently taken by the EM routines.

The sequence of events which take place in SETUP are the following. First, files are opened and the input is read in from file INDUCT94.IN. Constant variables are then initialized. The initial setup data is written to file INDUCT94.OUT.

Gridding is calculated using the subroutine POWER if needed. POWER calculates a stretched, non-linear mesh component if sheath resolution is desired. The mesh is orthogonal in cylindrical (r, z) coordinates. The input variables which determine the fluid routines mesh are: RMAX, RSHEATH, ZMAX, ZSHEATH, NR, NRI, NZ, AND NZI. Additional variables used to set up the EM routine mesh are: NREM and NZEM. The EM routine mesh is also orthogonal in cylindrical coordinates, but is strictly uniformly spaced. For the fluid routines, NR and NZ correspond respectively to the total number of r and z grid cells or cell centers. For the EM routines, nrem and nzem respectively give the total number of cell boundaries, with the number of cell centers being NREM - 1 and NZEM - 1. If RSHEATH or ZSHEATH is less than or equal to zero, then the fluid mesh is uniform, otherwise the mesh is broken up into two region is r (one from 0 to RMAX - RSHEATH and the second from RMAX - RSHEATH to RMAX) and three regions in z (one from 0 to ZSHEATH, the second from ZSHEATH to ZMAX - ZSHEATH, and third from ZMAX - ZSHEATH to ZMAX). The variables NRI and NZI give the number of cells within the surface regions which are of width RSHEATH and ZSHEATH. This leaves NR - NRI and NZ - 2 * NZI cells in the central grid region. The mesh is uniformly spaced in r in the interior region. The mesh matches across the radial position RSHEATH and is non-uniform from RSHEATH to RMAX. Similarly in z, the z mesh is uniform in the interior zone and non-uniform in the two surface regions. In addition to the grid which runs from 0–RMAX and 0–ZMAX, "ghost" cells are added exterior about the exterior for boundary condition usage. Arrays for cell centered quantities, including these "ghost" cells have dimensions 0:MRP1 in the r direction, and 0:MZP1 in the z direction. Arrays for variables evaluated at the cell boundaries, such as fluxes, have dimensions MRP1 in r and MZP1 in z. If large areas are setup with structures, including vacuum regions containing the rf coils, the region of the grid containing the plasma may be fairly small. To reduce the fluid routines computation, we limit the range of many of the do-loops to 1–NRP, and 1–NZP, where NRP and NZP respectively give the maximum radial and axial indices of the plasma grid cells.

After setting up the grid the DC potential field in calculated through a call to subroutine SETPOT. As the plasma starts out neutral, the initial field consists of a vacuum potential field given by fixed potential on internal and boundary structures. In INDUCT94 quite general internal and boundary structures can in principle be treated. Each structure is assigned a fixed voltage. Irregular shapes, such as the curved dome of a bell jar, are treated with stair steps boundaries. Such structures are said to be "masked". Currently only simple structures are allowed. Mask arrays are set by a call from SETPOT to the subroutine MASKER.

The subroutine MASKER sets the variables: nmask, mask, masktype, propmask, nmaskgap, nmaskcath, oldphimask, phimask, nmaskr, maskr, nmaskl, maskl, nmasku, masku, nmasdd, maskd, eps, epshr, epshz. These variables define the number of masked structures, the corresponding masked grid cells, the type of structure, the structure property, the mask number for the air gap and cathode (substrate holder), the structure potential, and the dielectric constants. In addition, information on cells adjacent to each side of the structure surface, either to the left, the right, above, or below is also set.

Depending upon the placement of variables ZWIN1, ZWIN2, RWIN1, RWIN2, ZCENTER, RCENTER, RCENTER1, RCENTER2, ZCATH, RCATH, ZBLOCK, and RBLOCK various structures may be generated which describe the reactor structure (see Figure 5). For the sample test problem, we have set RWIN1, ZCENTER, and RCENTER to be outside of the modeling grid which goes from 0–RMAX AND 0–ZMAX. This leads to a very simple structure geometry (see Figure 6), which correspond to the reactor window region above the plasma and the substrate holder (or cathode).

To solve for the DC fields, SETPOT calls subroutine D03EBF which solves Poisson's equation using the Strongly ImPlicit (SIP) scheme. After calculating the DC potential and saving it is variable phi, SETPOT calls subroutine EFIELD to calculate the DC electric fields er and ez. These variables give the fields at cell boundaries.

Further variable initialization is conducted after the DC fields are generated. If the simulation is to be restarted from a previous calculation the variable istart must be greater than 1. If multiple restart (and graphic post-processing) files are to be generated during a simulation then the variable multidump must be set true. If restarting from a multiple restart calculation, then istart should correspond to the restart file number. When restarting, output is appended to the old INDUCT94.OUT file which must exist or an error message is generated. When restarting a simulation, the variable multidump should not be changed.

If the simulation is a continuation run from a previous simulation then densities and other fluid variables are initialized from the restart files, otherwise a call is make to the subroutine SETHYDRO. The subroutine SETHYDRO sets the initial fluid model variables.

The time step to be used is either the restored time step for a restart from dumpfile run, or is the minimum of the input initial time step dt0 and the Courant condition time step

$$\mathrm{DTC} = \mathrm{min}_{i,j}\left\{\frac{\mathrm{DRB}(i)}{\mathrm{abs}[\mathrm{VMR}(i,j)]}, \frac{\mathrm{DBZ}(j)}{\mathrm{abs}[\mathrm{VMR}(i,j)]}\right\}, \tag{27}$$

where (i, j) run over the fluid model r and z grid and the velocities refer to the electron drift velocities.

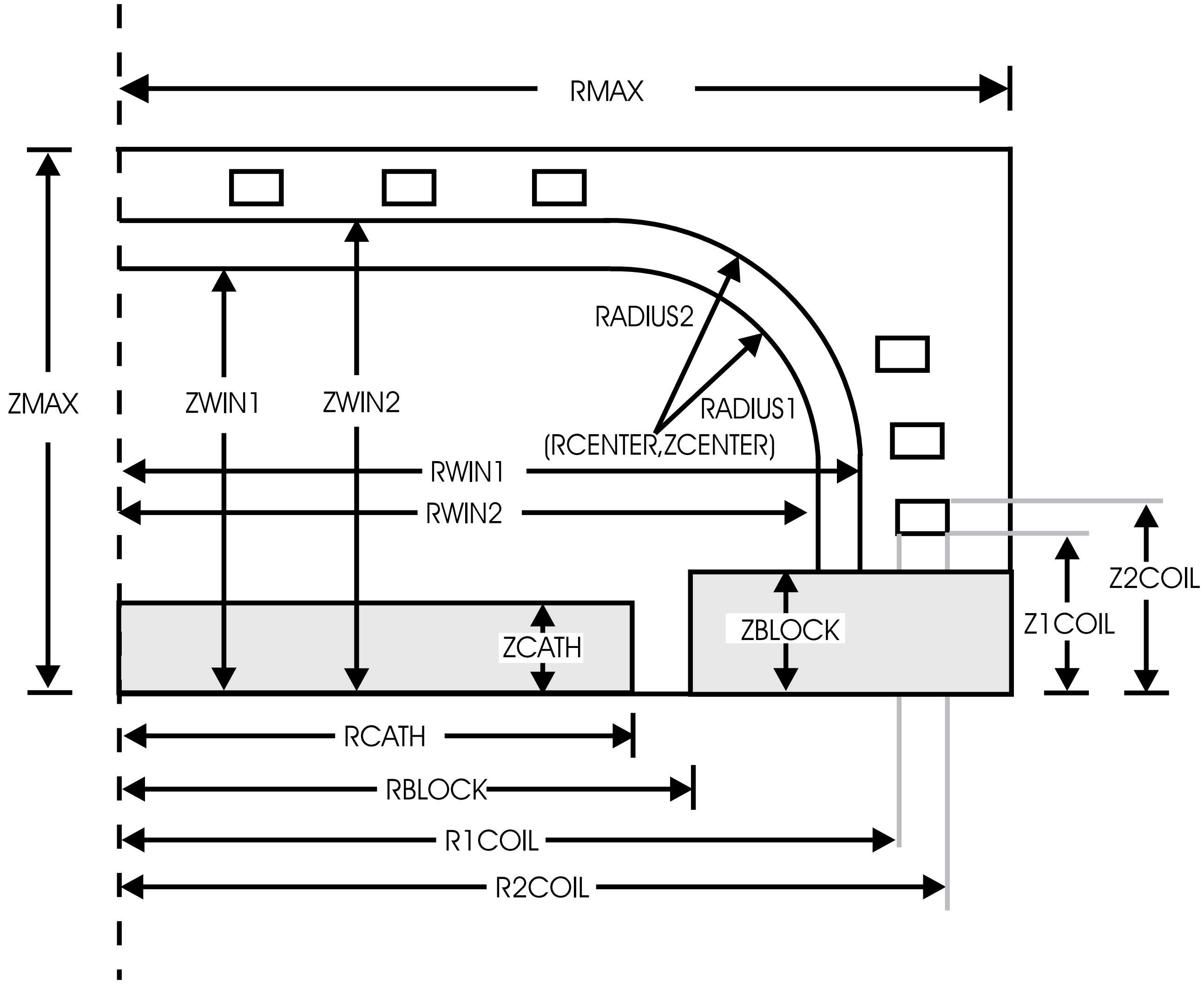


Figure 5: Geometry of possible INDUCT94 structures.

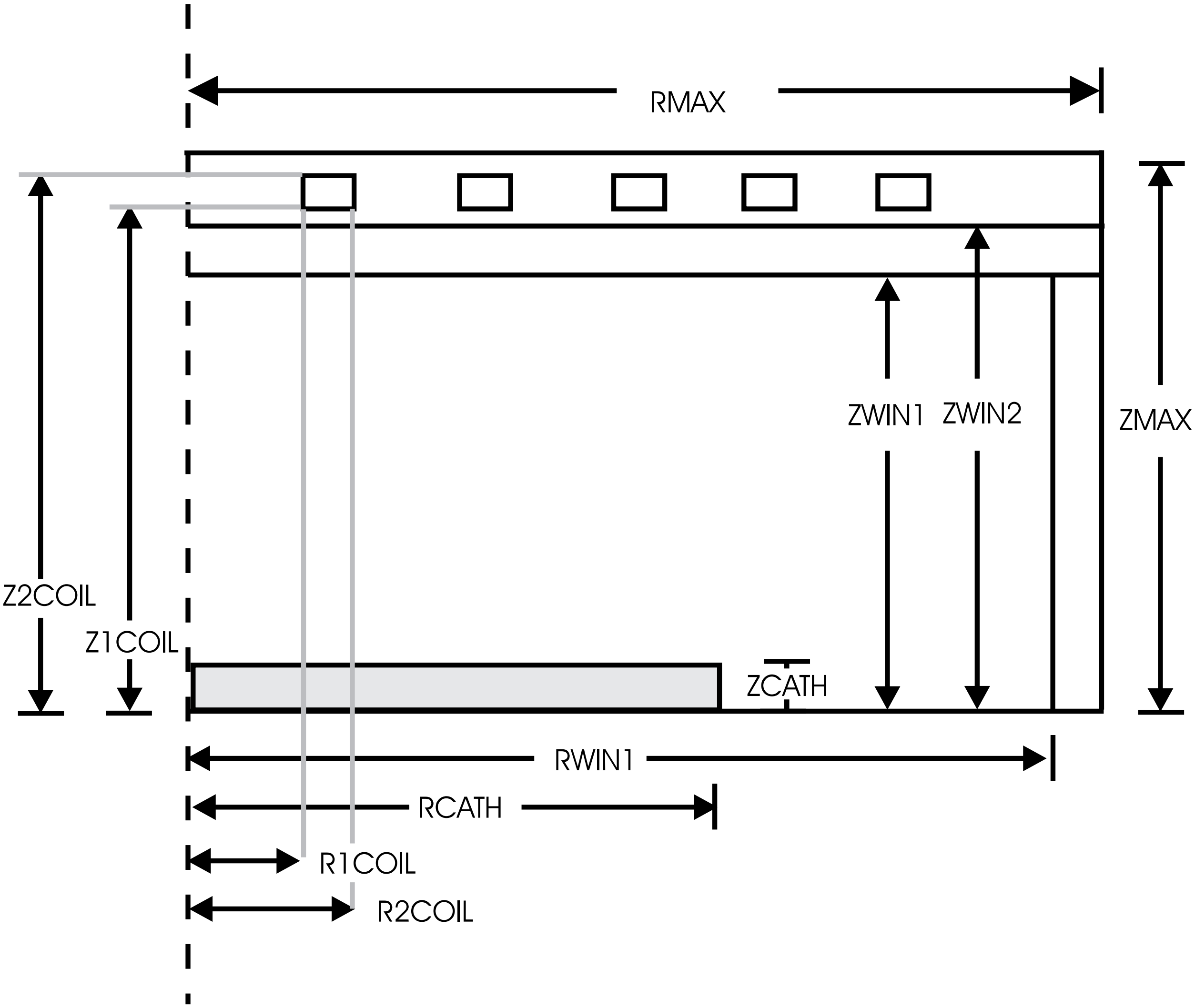


Figure 6: Geometry of INDUCT94 sample problem structures.

Upon determining the time step, if the simulation is not a restart run, then a call is make to the subroutine OUTPUT to save the initial data values. Next the subroutine TIMER is called, again only if this is not a restart calculation. For restart calculations, timing data is not accurately saved. For all simulations, whether restarted or not, the subroutine RATES is called to ensure that atomic physics rates consistent with current temperatures and densities are available. If the EM routines are to be called for non-uniform rf power deposition the subroutine GRIDMAP is called to initialize the interpolation variables between the fluid grid and the EM grid. Finally, if the simulation is not a restart run, and non-uniform rf power deposition is desired, then the subroutine IHEAT is called to determine the spatially varying rf heating rate. Note that these heating rates are currently computationally expensive to determine, so IHEAT is called sparingly. For a converged solution the frequency of calls to IHEAT is not important. We find recalculating the heating rates every 10–20 microseconds sufficient.

4 Sample Model

We consider here a sample modeling test problem for which we describe the input data used by INDUCT, and the output produced by INDUCT94. We also discuss how post-processing the data files can be done for detailed analysis. This sample problem is for Argon at 20 mTorr. The overall reactor is 20 cm in radius by 10 cm in height, with the plasma chamber having a radius of 15 cm and a 6 cm height (see Figure 6). Five inductive coils driven at 13.56 MHtz are placed on top of the plasma region. The rf inductive power absorbed is 100 W. No substrate bias is applied.

4.1 Sample Problem Input

We reproduce here the sample input file used for our example. This is a FORTRAN NAMELIST file. The format may vary slightly for different compilers.

```
c     INDUCT94 version 2.3                                    1/05/95
c     (c) The Regents of the University of California
c     Authors:  Peter Vitello, Richard A. Stewart, David B. Graves,
c               E. Fred Jaeger, and Lee A. Berry

 &input
  rmax      =   20.d0,
  zmax      =   10.d0,
  nr        =      60,
  nz        =      30,
  phir      =    0.d0,
  den0      =  1.0d11,
  mstep     = 5000000,
mcycle    = 5000000,
mstepp    = 5000000,
mcyclep   = 5000000,
msteppr   = 5000000,
mrf       =     200,
nrfstepmod=      20,
dt0       =  1.d-11,
mt        =  1.0d-6,
mtp       =  1.0d-7,
mtpr      =  1.0d-5,
iorder    =       1,
cour      =   1.0d0,
dtefactor =  1.02d0,
dtifactor =   0.1d0,
phiacc    =  1.0d-4,
phiacco   =  1.0d-4,
aparam    =   75.d0,
conres    =   0.1d0,
istart    =       0,
itmaxsip  =    2000,
itmaxsor  =    2000,
msip      =       1,
nspec     =       3,
negspec   =       0,
neutspec  =       1,
spect(2)  =       2,
spect(3)  =       3,
specl(2)  =   'Ar+',
specl(3)  =    'Ar',
specm(2)  =    40.0,
specm(3)  =    40.0,
specq(2)  =    1.d0,
specq(3)  =    0.d0,
fracinit(2) =  1.d0,
fracinit(3) =  1.d0,
nchem   = 7,
ic(1,1) = 1, ic(2,1) = 3, ic(3,1) = 2, ic(4,1) = 1, ic(5,1) = 1,
ic(1,2) = 1, ic(2,2) = 3, ic(3,2) = 3, ic(4,2) = 1, ic(5,2) = 0,
ic(1,3) = 1, ic(2,3) = 3, ic(3,3) = 3, ic(4,3) = 1, ic(5,3) = 0,
ic(1,4) = 1, ic(2,4) = 3, ic(3,4) = 3, ic(4,4) = 1, ic(5,4) = 0,
ic(1,5) = 1, ic(2,5) = 3, ic(3,5) = 3, ic(4,5) = 1, ic(5,5) = 0,
ic(1,6) = 1, ic(2,6) = 3, ic(3,6) = 3, ic(4,6) = 1, ic(5,6) = 0,
ic(1,7) = 1, ic(2,7) = 3, ic(3,7) = 3, ic(4,7) = 1, ic(5,7) = 0,
csectin(2,3) = 5.d-15,
tempi0       = 0.05d0,
tempe0       = 2.00d0,
localte      = .true.,
uniform      =.false.,
secondary(2,1) = 0.d0,
secondary(2,2) = 0.d0,
secondary(2,3) = 0.d0,
multidump    = .true.,
nrem = 40,
nzem = 46,
niter = 20,
epsit = 1.d-05,
lbias          = .false.,
coilcapcoup    = .false.,
coilfreq       = 13.56d6,
biasfreq       = 13.56d6,
biascapac      = 6.28d-10,
biasresistance = 3.d0,
ncoil   = 5,
iin     = (20.d0,0.0d0),
prf     = 100.d0,
zout    = (0.0d3,0.0d0),
ifeed   = 1,
zwin1   = 6.00d0,
zwin2   = 7.50d0,
xkwin   = 3.75d0,
rwin1   = 15.0d0,
rwin2   = 16.2d0,
rcath   = 10.0d0,
zcath   = 00.0d0,
rblock  = 100.d0,
zblock  = 00.0d0,
acbias  = (0.0d0,0.0d0),
dcbias  = 00.0d0,
rcenter = 100.000,
zcenter = 100.000,
radius1 = 100.000,
radius2 = 100.000,
r1coil  = 1.2d0, 3.2d0, 5.2d0, 7.2d0,  9.2d0, 0.0, 0.0, 0.0, 0.0,
r2coil  = 2.8d0, 4.8d0, 6.8d0, 8.8d0, 10.8d0, 0.0, 0.0, 0.0, 0.0,
  z1coil  = 7.5d0, 7.5d0, 7.5d0, 7.5d0,  7.5d0, 0.0, 0.0, 0.0, 0.0,
  z2coil  = 8.1d0, 8.1d0, 8.1d0, 8.1d0,  8.1d0, 0.0, 0.0, 0.0, 0.0,
 /
 &neutral
  pressure  =     20.d0,
  nchems    =         0,
  nreacts   =         0,
  nprods    =         0,
  ics(1,1)  =         0,
  gammac(1) =   0.000d0,
  ratest    =         0,
  ratesl    =       ' ',
  tempn0    =   0.025d0,
  flow(3)   =   100.0d0,
  gammar(3) =    0.01d0,
  gammai(3) =   1.000d0,
  accel     =     1.0d2,
  freact(1)  ='./argon/ioniz.k',
  freact(2)  ='./argon/excitmeta.k',
  freact(3)  ='./argon/elastic.k',
  freact(4)  ='./argon/excit3d132.k',
  freact(5)  ='./argon/excit3d32.k',
  freact(6)  ='./argon/excit5d12.k',
  freact(7)  ='./argon/excit5s32.k',
 /
```

4.2 Sample Output

The output file for the sample problem follows:

```
*****FLUID ROUTINES INPUT PARAMETERS*****

istart =         0, itmaxsip=       2000, itmaxsor =    2000, msip  = 1

rmax   = 2.000E+01, rsheath =  0.000E+00
zmax   = 1.000E+01, zsheath =  0.000E+00

nr     = 60
nz     = 30
nri    =   0
nzi    =   0
```

```
phir   = 0.00E+00, den0   = 1.00E+11, pressure =  2.000E+01
tempi0 = 5.00E-02, tempe0 = 2.00E+00, tempn0   =  2.500E-02

localte= T, localti    = F, uniform  = F, lzerone = F, multidump = T

mstep  = 5000000, mstepp  =  5000000, msteppr   =   5000000
mcycle = 5000000, mcyclep =  5000000, nrfstepmod=       20
mrf    =       200

dt0    = 1.0E-11, mt      = 1.0E-06, mtp      = 1.00E-07, mtpr      = 1.0E-05

iorder =       1, cour    = 1.0E+00, dtefactor= 1.02E+00, dtifactor = 1.0E-01
phiacc = 1.0E-04, phiacco = 1.0E-04, aparam   = 7.50E+01, conres    = 1.0E-01

nspec  =          3, negspec =          0, neutspec =         1
nchem  =          7
nchems =          0

Volume Rates
rate  nreact    nprod    ic
  1      2        3      1    3    2    1    1
  2      2        2      1    3    3    1
  3      2        2      1    3    3    1
  4      2        2      1    3    3    1
  5      2        2      1    3    3    1
  6      2        2      1    3    3    1
  7      2        2      1    3    3    1

rate    eloss
  1   1.560E+01
  2   1.160E+01
  3   0.000E+00
  4   1.430E+01
  5   1.415E+01
  6   1.426E+01
  7   1.409E+01

rate  type            label
  1     1   Ar Ionization   e + Ar --> Ar+ + e + e
  2     3   Ar excitation to metastable   e + Ar --> Ar* + e --> Ar + e
  3     2   Ar elastic scattering   e + Ar --> e + Ar
  4     3   Ar excitation to 3d^1 3/2   e + Ar --> Ar** + e --> Ar + e
  5     3   Ar excitation to 3d 3/2   e + Ar --> Ar** + e --> Ar + e
  6     3   Ar excitation to 3d 1/2   e + Ar --> Ar** + e --> Ar + e
  7     3   Ar excitation to 5s 3/2   e + Ar --> Ar** + e --> Ar + e

                    Species
     1          Electron
     2          Ar+
     3          Ar

spec    charge    mass    type fracinit    flow     gammar    gammai
  1  -1.000E+00 9.110E-28   1 1.000E+00 0.000E+00 0.000E+00 0.000E+00
  2   1.000E+00 6.690E-23   2 1.000E+00 0.000E+00 0.000E+00 0.000E+00
  3   0.000E+00 6.690E-23   3 1.000E+00 1.000E+02 1.000E-02 1.000E+00

accel = 1.000E+02


*****EM SOLVER INPUT PARAMETERS*****

nrem     =         40, nzem    =         46
ncoil    =          5, ifeed   =          1, coilfreq     =  1.356E+07

prf      =  1.000E+02, iin     = ( 2.000E+01, 0.000E+00)
                       zout    = ( 0.000E+00, 0.000E+00)

niter    =        20
zwin1    = 6.00E+00, zwin2   = 7.50E+00, rwin1   = 1.50E+01, rwin2   = 1.62E+01
xkwin    = 3.75E+00, epsit   = 1.00E-05
rcath    = 1.00E+01, zcath   = 0.00E+00, xncath  = 8.34E+13

dcbias   = 0.00E+00, acbias    = ( 0.000E+00, 0.000E+00)
biasfreq = 1.356E+07, biascapac =   6.280E-10

rdop     = 1.00E+02, zdop    = 1.00E+02, sigdop  = 1.00E+01
rcenter  = 1.00E+02, zcenter = 1.00E+02, radius1 = 1.00E+02, radius2 = 1.00E+02

   r1coil        r2coil        z1coil        z2coil
 1.200E+00     2.800E+00     7.500E+00     8.100E+00
 3.200E+00     4.800E+00     7.500E+00     8.100E+00
 5.200E+00     6.800E+00     7.500E+00     8.100E+00
 7.200E+00     8.800E+00     7.500E+00     8.100E+00
 9.200E+00     1.080E+01     7.500E+00     8.100E+00


          END OF INPUT DATA

************************************************************
************************************************************
************************************************************
************************************************************




##########

 dumpfile = 00 , running time (min) = .47533
 t      = 0.000E+00, istep =         0, dt     = 1.000E-11, dte    = 1.020E-11
 dtrf   = 0.000E+00, dtc   = 3.333E+19, dtchem = 0.000E+00, itcoun =         1
 phimax = 1.000E-20
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  9.978E+10   0   0  1.477E+14   0.000E+00   0.000E+00   0.000E+00   0.000E+00
   2  9.978E+10   0   0  1.477E+14   0.000E+00   0.000E+00   0.000E+00   0.000E+00
   3  6.600E+14   0   0  2.644E+18   0.000E+00   0.000E+00   0.000E+00   0.000E+00

 spec  frch_h      frch_hn     (i,j)   frch_i      frch_in     (i,j)   tempmax   (i,j)
   1  0.000E+00   0.000E+00   1  10  0.000E+00   0.000E+00   0   0  2.000E+00  1   2
   2  0.000E+00   0.000E+00   1  10  0.000E+00   0.000E+00   0   0  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  0.000E+00   0.000E+00   0   0  2.500E-02  1   2

 spec  volume gain  wall ion gain   wall loss        flow     wall chem gain
   3   0.00000E+00  0.00000E+00    0.00000E+00   4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den      frac        -frac*den   dden-frac*den   dden
   3   6.60000E+14  0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00E+00

  pabs               iin                     icoil                      vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 01 , running time (min) = 1.7687
 t      = 1.003E-07, istep =       268, dt     = 1.943E-09, dte    = 1.981E-09
 dtrf   = 0.000E+00, dtc    = 1.943E-09, dtchem = 2.896E-06, itcoun =         6
 phimax = 1.906E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  9.988E+10   1  10  1.478E+14   2.932E+18  -1.667E+18   1.936E+11  -1.350E+11
   2  9.988E+10   1  10  1.478E+14   2.932E+18  -1.974E+18   1.936E+11   1.004E+11
   3  6.657E+14   1   2  2.666E+18  -2.932E+18  -5.341E-18  -1.936E+11   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  3.579E-04   4.538E+09  45  18  6.707E-05   2.483E+10  24  18  2.407E+00 24  18
   2  4.035E-04   4.655E+09  45  18  6.692E-05   2.571E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  5.421E-09   6.657E+14  17  12  2.500E-02  1   2

 spec  volume gain  wall ion gain   wall loss        flow     wall chem gain
   3  -2.93165E+18  1.97411E+18   -1.26570E+20  4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den     frac        -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65692E+14 -6.30324E-06   4.19602E+09  -2.11519E-01 -4.1960E+09

  pabs               iin                     icoil                    vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 02 , running time (min) = 1.9590
 t      = 2.009E-07, istep =       319, dt     = 1.995E-09, dte    = 2.034E-09
 dtrf   = 0.000E+00, dtc    = 1.995E-09, dtchem = 1.623E-06, itcoun =         6
 phimax = 1.773E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0
 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  1.001E+11   1  10  1.479E+14   5.655E+18  -3.833E+18   6.236E+11  -4.176E+11
   2  1.001E+11   1  10  1.480E+14   5.655E+18  -3.700E+18   6.236E+11   3.904E+11
   3  6.657E+14   1   2  2.666E+18  -5.655E+18  -5.341E-18  -6.236E+11   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  8.015E-04   4.413E+09  45  18  1.229E-04   2.537E+10  23  18  2.600E+00 23  18
   2  7.972E-04   4.522E+09  45  18  1.227E-04   2.542E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  1.077E-08   6.657E+14  16  12  2.500E-02  1   2

 spec  volume gain  wall ion gain   wall loss        flow     wall chem gain
   3  -5.65496E+18  3.70023E+18   -1.26570E+20   4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den     frac        -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65692E+14 -6.54682E-06   4.35816E+09  -9.30415E-01 -4.3581E+09

  pabs               iin                      icoil                    vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 03 , running time (min) = 2.0927
 t      = 3.012E-07, istep =       369, dt     = 2.012E-09, dte    = 2.052E-09
 dtrf   = 0.000E+00, dtc   = 2.012E-09, dtchem = 1.130E-06, itcoun =        7
 phimax = 1.850E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  1.004E+11   1  10  1.482E+14   8.542E+18  -4.996E+18   1.340E+12  -8.728E+11
   2  1.005E+11   1  10  1.482E+14   8.542E+18  -5.113E+18   1.340E+12   8.361E+11
   3  6.657E+14   1   2  2.666E+18  -8.542E+18  -5.341E-18  -1.340E+12   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  1.135E-03   4.216E+09  45  18  1.780E-04   2.495E+10  23  18  2.735E+00 23  18
   2  1.125E-03   4.323E+09  45  18  1.777E-04   2.499E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  1.633E-08   6.657E+14  16  12  2.500E-02  1   2
 spec  volume gain  wall ion gain   wall loss       flow     wall chem gain
   3  -8.54192E+18  5.11333E+18   -1.26570E+20   4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den     frac       -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65692E+14 -6.71580E-06   4.47065E+09  -1.58315E-01 -4.4706E+09

  pabs               iin                     icoil                    vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 04 , running time (min) = 2.1758
 t      = 4.022E-07, istep =       419, dt     = 2.027E-09, dte    = 2.068E-09
 dtrf   = 0.000E+00, dtc   = 2.027E-09, dtchem = 9.101E-07, itcoun =         5
 phimax = 1.867E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  1.010E+11   1  10  1.486E+14   1.102E+19  -6.142E+18   2.335E+12  -1.438E+12
   2  1.010E+11   1  10  1.487E+14   1.102E+19  -6.227E+18   2.335E+12   1.413E+12
   3  6.657E+14   1   2  2.666E+18  -1.102E+19  -5.341E-18  -2.335E+12   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  1.405E-03   3.974E+09  45  18  2.227E-04   2.442E+10  23  18  2.824E+00 23  18
   2  1.382E-03   4.078E+09  45  18  2.224E-04   2.446E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  2.122E-08   6.657E+14  14  12  2.500E-02  1   2

 spec  volume gain  wall ion gain   wall loss       flow     wall chem gain
   3  -1.10159E+19  6.22703E+18   -1.26570E+20   4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den     frac       -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65691E+14 -6.86896E-06   4.57261E+09  -2.83874E-01 -4.5726E+09

  pabs               iin                     icoil                    vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 05 , running time (min) = 2.2590
 t      = 5.038E-07, istep =       469, dt     = 2.038E-09, dte    = 2.079E-09
 dtrf   = 0.000E+00, dtc    = 2.038E-09, dtchem = 8.043E-07, itcoun =          5
 phimax = 1.865E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  1.016E+11   1  10  1.492E+14   1.275E+19  -7.064E+18   3.552E+12  -2.113E+12
   2  1.016E+11   1  10  1.492E+14   1.275E+19  -7.016E+18   3.552E+12   2.089E+12
   3  6.657E+14   1   2  2.666E+18  -1.275E+19  -5.341E-18  -3.552E+12   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  1.595E-03   3.708E+09  45  18  2.533E-04   2.384E+10  23  18  2.877E+00 23  18
   2  1.564E-03   3.810E+09  45  18  2.530E-04   2.387E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  2.488E-08   6.657E+14  14  12  2.500E-02  1   2

 spec  volume gain  wall ion gain   wall loss        flow     wall chem gain
   3  -1.27549E+19  7.01554E+18   -1.26570E+20   4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den     frac        -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65691E+14 -6.97749E-06   4.64485E+09   1.01597E+00 -4.6448E+09

  pabs                iin                     icoil                    vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 06 , running time (min) = 2.3388
 t      = 6.059E-07, istep =       519, dt     = 2.037E-09, dte    = 2.078E-09
 dtrf   = 0.000E+00, dtc    = 2.037E-09, dtchem = 7.545E-07, itcoun =          4
 phimax = 1.877E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  1.024E+11   1  10  1.498E+14   1.384E+19  -7.532E+18   4.914E+12  -2.862E+12
   2  1.024E+11   1  10  1.498E+14   1.384E+19  -7.519E+18   4.914E+12   2.833E+12
   3  6.657E+14   1   2  2.666E+18  -1.384E+19  -5.341E-18  -4.914E+12   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  1.707E-03   3.436E+09  45  18  2.700E-04   2.323E+10  23  18  2.906E+00 23  18
   2  1.668E-03   3.536E+09  45  18  2.696E-04   2.327E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  2.707E-08   6.657E+14  13  11  2.500E-02  1   2

 spec  volume gain  wall ion gain   wall loss        flow     wall chem gain
   3  -1.38425E+19  7.51875E+18   -1.26570E+20   4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den     frac        -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65691E+14 -7.01974E-06   4.67298E+09  -8.59956E-02 -4.6729E+09

  pabs               iin                    icoil                    vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 07 , running time (min) = 2.4195
 t      = 7.074E-07, istep =       569, dt     = 2.025E-09, dte    = 2.066E-09
 dtrf   = 0.000E+00, dtc   = 2.025E-09, dtchem = 7.278E-07, itcoun =        4
 phimax = 1.894E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  1.032E+11   1  10  1.505E+14   1.449E+19  -7.718E+18   6.356E+12  -3.637E+12
   2  1.032E+11   1  10  1.505E+14   1.449E+19  -7.801E+18   6.356E+12   3.613E+12
   3  6.657E+14   1   2  2.666E+18  -1.449E+19  -5.341E-18  -6.356E+12   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  1.749E-03   3.174E+09  45  18  2.783E-04   2.263E+10  23  18  2.921E+00 23  18
   2  1.705E-03   3.273E+09  45  18  2.779E-04   2.267E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  2.824E-08   6.657E+14  13  11  2.500E-02  1   2

 spec  volume gain  wall ion gain   wall loss        flow     wall chem gain
   3  -1.44888E+19  7.80125E+18   -1.26570E+20   4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den     frac        -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65691E+14 -7.00647E-06   4.66414E+09  -1.69203E-01 -4.6641E+09

  pabs               iin                     icoil                    vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 08 , running time (min) = 2.5018
 t      = 8.084E-07, istep =       619, dt     = 2.015E-09, dte    = 2.056E-09
 dtrf   = 0.000E+00, dtc   = 2.015E-09, dtchem = 7.159E-07, itcoun =         5
 phimax = 1.890E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  1.040E+11   1  10  1.512E+14   1.485E+19  -7.928E+18   7.840E+12  -4.427E+12
   2  1.040E+11   1  10  1.512E+14   1.485E+19  -7.922E+18   7.840E+12   4.409E+12
   3  6.657E+14   1   2  2.666E+18  -1.485E+19  -5.341E-18  -7.840E+12   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  1.749E-03   2.930E+09  45  18  2.815E-04   2.207E+10  23  18  2.929E+00 23  18
   2  1.698E-03   3.027E+09  45  18  2.811E-04   2.210E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  2.888E-08   6.657E+14  13  11  2.500E-02  1   2

 spec  volume gain  wall ion gain   wall loss        flow     wall chem gain
   3  -1.48455E+19  7.92205E+18   -1.26570E+20   4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den     frac        -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65691E+14 -6.98940E-06   4.65279E+09  -1.82919E-01 -4.6527E+09

  pabs               iin                     icoil                    vcoil
```

```
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 09 , running time (min) = 2.5842
 t      = 9.090E-07, istep =       669, dt     = 2.007E-09, dte    = 2.048E-09
 dtrf   = 0.000E+00, dtc   = 2.007E-09, dtchem = 7.124E-07, itcoun =         4
 phimax = 1.889E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  1.048E+11   1  10  1.519E+14   1.502E+19  -8.001E+18   9.343E+12  -5.231E+12
   2  1.048E+11   1  10  1.519E+14   1.502E+19  -7.932E+18   9.343E+12   5.206E+12
   3  6.657E+14   1   2  2.666E+18  -1.502E+19  -5.341E-18  -9.343E+12   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  1.717E-03   2.707E+09  45  18  2.817E-04   2.155E+10  23  18  2.931E+00 23  18
   2  1.661E-03   2.804E+09  45  18  2.813E-04   2.159E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  2.919E-08   6.657E+14  13  11  2.500E-02  1   2

 spec  volume gain  wall ion gain    wall loss        flow     wall chem gain
   3  -1.50163E+19  7.93187E+18    -1.26570E+20   4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den      frac        -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65691E+14 -6.97331E-06   4.64207E+09  -1.08996E-01 -4.6420E+09

  pabs                iin                      icoil                    vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                    ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                    ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                    ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                    ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


##########

 dumpfile = 10 , running time (min) = 2.6580
 t      = 1.001E-06, istep =       715, dt     = 2.001E-09, dte    = 2.041E-09
 dtrf   = 0.000E+00, dtc   = 2.001E-09, dtchem = 7.123E-07, itcoun =        4
 phimax = 1.903E+01
 lbias =      F, ncycle =     0,   rfsteps/cycle=     0

 spec   peak      (i,j)     qvol       ionvol   currmaskall  ionvoltot    qmaskall
   1  1.055E+11   1  10  1.525E+14   1.511E+19  -7.858E+18   1.073E+13  -5.962E+12
   2  1.055E+11   1  10  1.525E+14   1.511E+19  -7.884E+18   1.073E+13   5.936E+12
   3  6.657E+14   1   2  2.666E+18  -1.511E+19  -5.341E-18  -1.073E+13   1.000E-20

 spec  frch_h      frch_hn    (i,j)   frch_i      frch_in    (i,j)   tempmax   (i,j)
   1  1.665E-03   2.521E+09  45  18  2.809E-04   2.113E+10  23  18  2.931E+00 23  18
   2  1.608E-03   2.618E+09  45  18  2.804E-04   2.116E+10  23  18  5.000E-02  1   2
   3  0.000E+00   0.000E+00   1   2  2.935E-08   6.657E+14  13  11  2.500E-02  1   2

 spec  volume gain  wall ion gain  wall loss       flow     wall chem gain
   3  -1.51077E+19  7.88434E+18  -1.26570E+20  4.10280E+19   0.00000E+00

 spec  qvol/volume   actual den     frac        -frac*den   dden-frac*den   dden
   3   6.65696E+14  6.65691E+14 -6.96224E-06   4.63470E+09  -9.11991E-01 -4.6347E+09

  pabs              iin                    icoil                    vcoil
1.000E+02  ( 6.812E+00, 0.000E+00)  ( 6.954E+00, 3.368E-03)  ( 2.443E+01,-1.033E+03)
                                     ( 7.340E+00, 1.259E-02)  ( 2.256E+01,-9.488E+02)
                                     ( 7.891E+00, 2.576E-02)  ( 1.835E+01,-7.786E+02)
                                     ( 8.408E+00, 3.792E-02)  ( 1.180E+01,-5.091E+02)
                                     ( 8.727E+00, 4.540E-02)  ( 4.025E+00,-1.744E+02)


 time (sec) =   1.60E+02
```

The output in INDUCT94.OUT begins with a listing of the input parameters used for the simulation. The calculated results begins with the line:

```
 dumpfile = 00 , running time (min) = 1.2778
```

which give the initial dumpfile number and the running time since the calculation began. If the model were restarted from previous dumpfile data (chosen by the value of the input variable ISTART), then the value for the dumpfile would reflect the value of ISTART. The next lines in INDUCT94.OUT give current values for time, time step, maximum potential, and data pertaining to time stepping and convergence of the Poisson solver.

The next data section gives for each species at the current time step:

spec: The particle species.

peak: The peak particle density [cm^{-3}].

(i, j): The corresponding radial and axial grid cell indices.

qvol: The volume integral of the particle density.

ionvol: The volume integrated rate of change due to chemistry [s^{-1}].

currmaskall: The subface integrated particle current [s^{-1}].

ionvoltot: The time integral of ionvol.

qmaskall: The time integral of currmaskall.

A negative value for ionvol, currmaskall, ionvoltot, or qmaskall implies a loss of this value of particles.

Results are then printed for the following quantities for each species:

spec: The particle species.

frch_h: The absolute magnitude of largest fractional changes due to fluid dynamics.

frch_hn: The particle density at the position frch_h peaks [cm^{-3}].

(i, j): The radial and axial index at the position frch_h peaks.

frch_i: The absolute magnitude of largest fractional changes due to chemistry.

frch_in: The particle density at the position frch_i peaks [cm^{-3}].

(i, j): The radial and axial index at the position frch_i peaks.

tempmax: The maximum temperature [eV].

(i, j): The radial and axial index corresponding to tempmax.

Neutral chemistry data follows the fractional change data:

spec: The neutral particle species.

volume gain: The volume integrated rate of change due to chemistry [s^{-1}]. This is equal to ionvol.

wall ion gain: The total gain in particles due to ion recombination on the walls [s^{-1}].

wall loss: The total loss due to wall sticking [s^{-1}].

flow: The total inflow of particles [s^{-1}]

wall chem gain: The total gain in particles due to chemical reactions on the walls [s^{-1}].

qvol/volume: Average neutral density after gains and losses [cm^{-3}].

actual den: Steady state pressure density desired [cm^{-3}].

frac: Fractional change needed to scale the average density to the constant value.

-frac*den: The change in the averaged density from the scaling [cm^{-3}].

dden-frac*den: The difference between the average change in density from gains and losses and the change in averaged density from the scaling [cm^{-3}].

dden: The average change in density from gains and losses [cm^{-3}].

The final data group at each output cycle gives information concerning the inducting rf coils. The variables printed are:

pabs: Inductive rf power absorbed by the plasma [W].

iin: Complex initial coil current [A].

icoil: Complex current at each coil [A].

vcoil: Complex voltage on each coil [V].

4.3 Output Postprocessing

Modeling results can be post-processed and graphed using the commercial interactive graphic program IDL or PVWAVE. A sample script file, WAVE.PLT, is provided with when run will automatically produce postscript plot files for electron density, potential, electron temperature, and rf power deposited for the 10th data dump (see Figures 7, 8, 9, 10. The text of WAVE.PLT follows.

```
; Grid dimension declaration
i=60
j=30

; Variable declarations
den = dblarr(i,j,3)
phi = dblarr(i,j)
temp = dblarr(i,j,3)
fluxr = dblarr(i+1,j+1,3)
fluxz = dblarr(i+1,j+1,3)
wdot = dblarr(i,j)
rffield = dblarr(i,j,3)
```

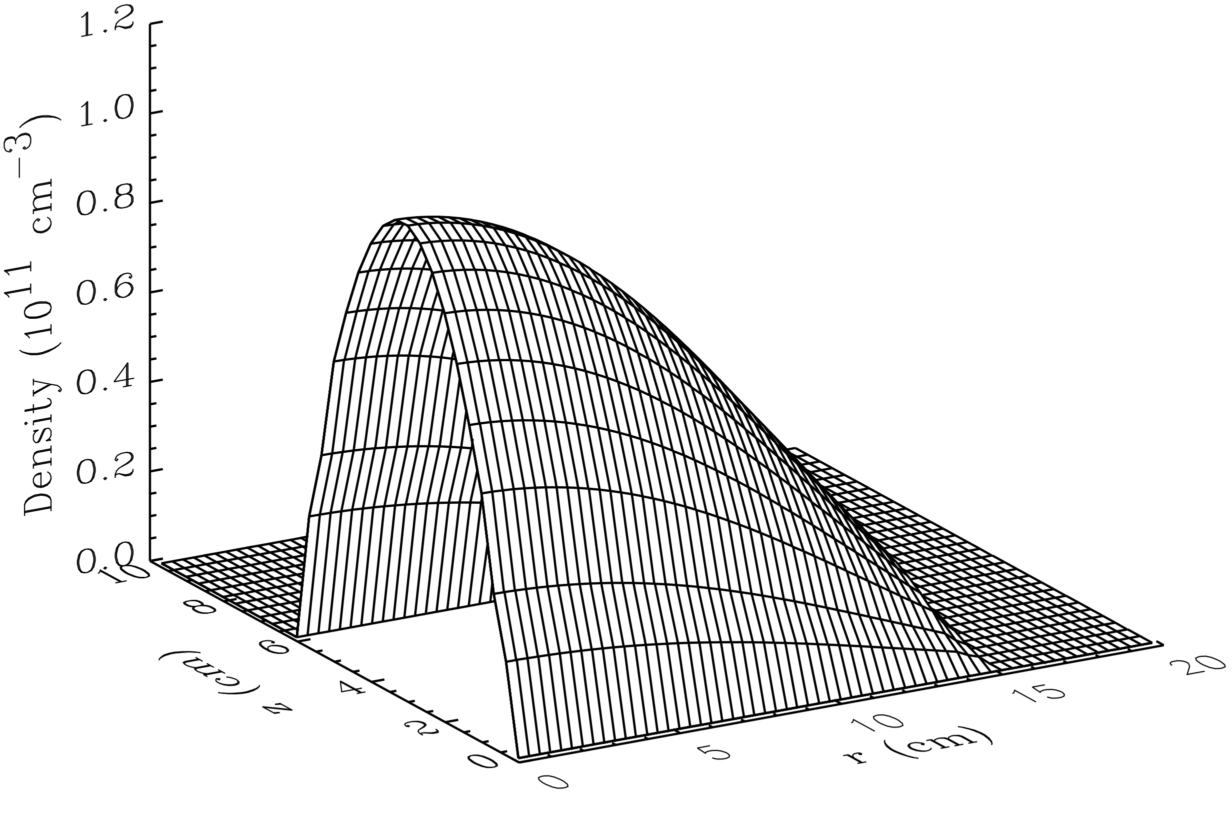


Figure 7: Electron density for sample problem.

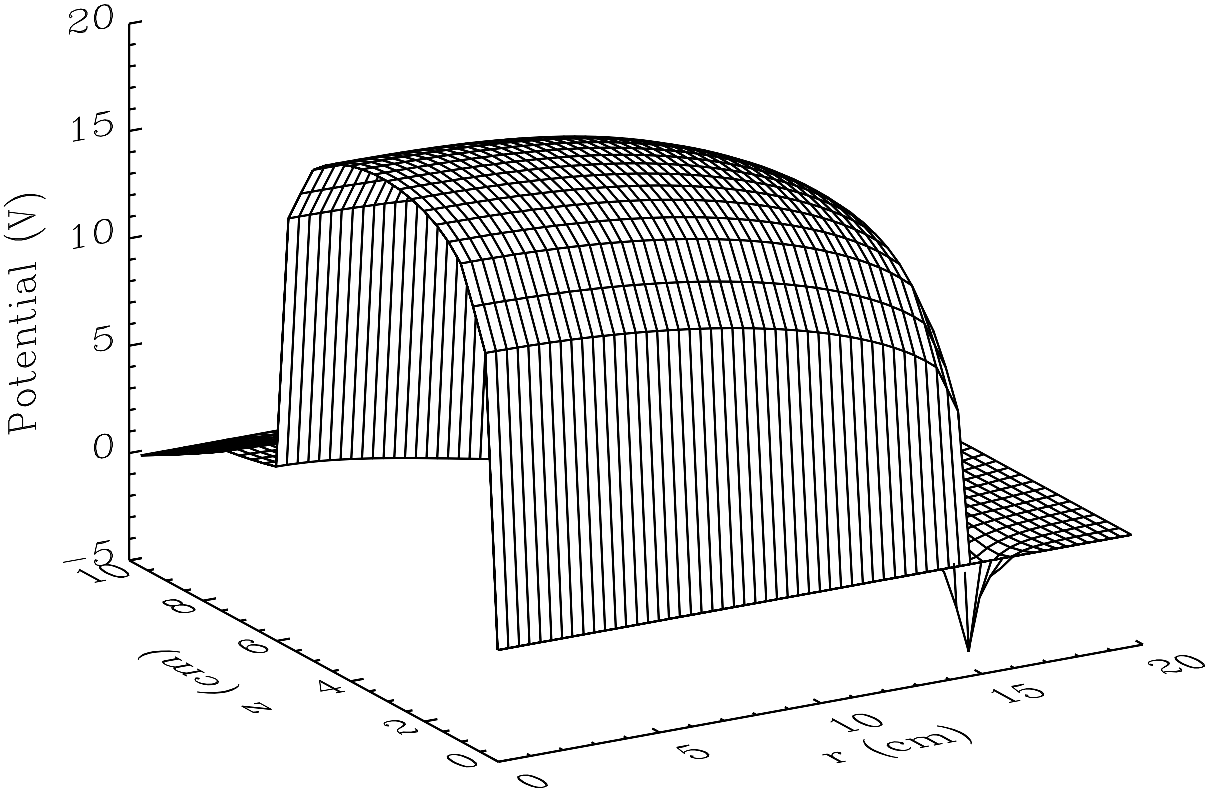


Figure 8: Potential for sample problem.

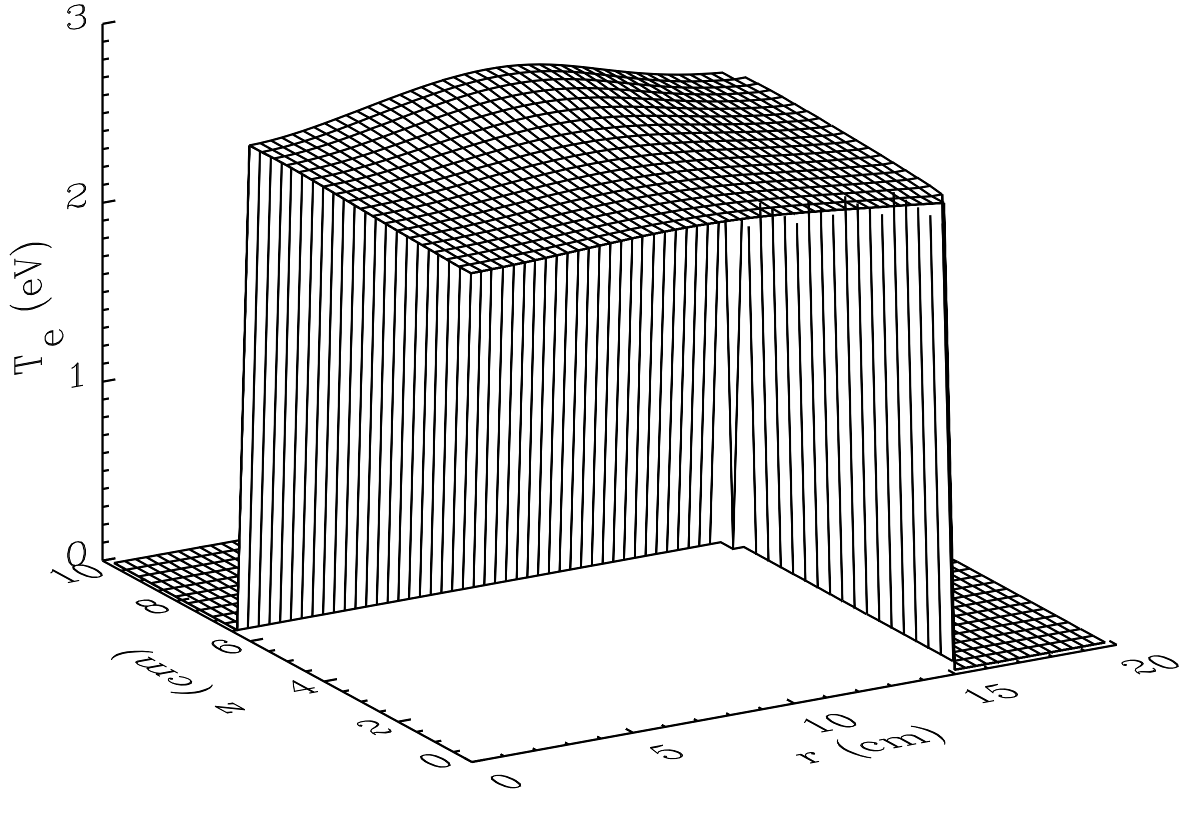


Figure 9: Electron temperature for sample problem.

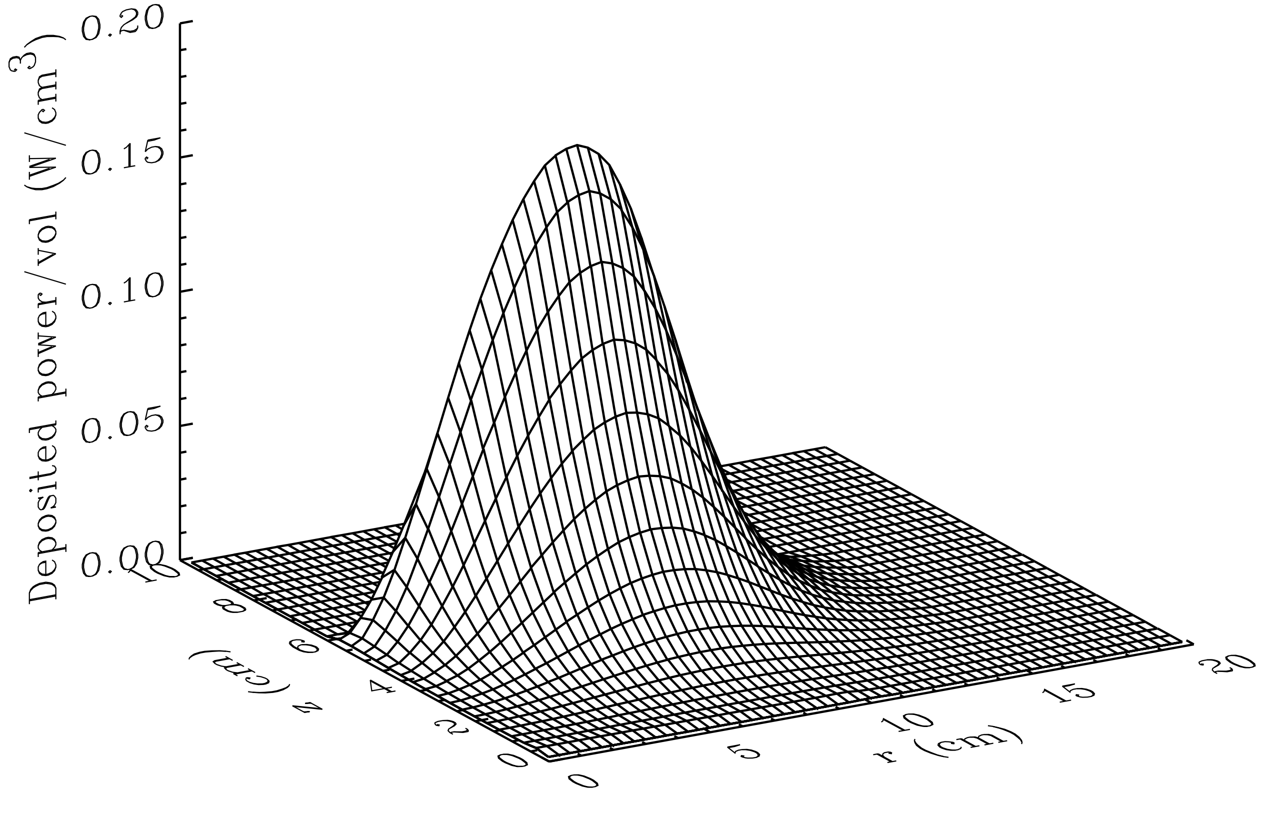


Figure 10: Inductive rf power deposition profile for sample problem.

```
; Grid array declarations
rc=dblarr(i)
zc=dblarr(j)
rb=dblarr(i+1)
zb=dblarr(j+1)

; Define input file names
files=strarr(4)
files(0)=’den10’
files(1)=’phi10’
files(2)=’tem10’
files(3)=’pow10’

; Read in data
openu,1,files(0),/f77_u
readu,1, den,rc,rb,zc,zb
close,1
openu,1,files(1),/f77_u
readu,1, phi
close,1
openu,1,files(2),/f77_u
readu,1, temp
close,1
openu,1,files(3),/f77_u
readu,1, wdot
close,1

; set default plot parameters
!P.Font=-1
!p.charsize=1.5
!p.charthick=1.5
!x.charsize=1.5
!y.charsize=1.5
!z.charsize=1.5
!y.minor=4
set_plot,’ps’
device,bits_per_pixel=8,/helvetica,/bold,/portrait
!p.thick=3
!x.thick=3
!y.thick=3
!z.thick=3
```

```
; plot density
device,filename='den.ps'
surface,den(*,*,0)/1.e11,rc,zc,xtit='!6r (cm)',ytit='!6z (cm)',
   ztit='!6Density (10!a11!n cm!a-3!n)'
device,/close

; plot potential
device,filename='phi.ps'
surface,phi(*,*),rc,zc,xtit='!6r (cm)',ytit='!6z (cm)',ztit='!6Potential (V)'
device,/close

; plot electron temperature
device,filename='temp.ps'
surface,temp(*,*,0),rc,zc,xtit='!6r (cm)',ytit='!6z (cm)',ztit='!6T!be!n (eV)'
device,/close

; plot power/volume deposited
device,filename='wdot.ps'
surface,wdot(*,*)/1.e7,rc,zc,xtit='!6r (cm)',ytit='!6z (cm)',
   ztit='!6Deposited power/vol (W/cm!a3!n)'
device,/close

exit
```

References

[1] M. A. Lieberman and R. A. Gottscho, *Physics of Thin Films*, Vol. 18, ed. M. Francombe and J. Vossen, Academic Press (1994).

[2] J. Hopwood, C. R. Guarnieri, S. J. Whitehair, and J. J. Cuomo, *J. Vac. Sci. Technol. A*, **11**, 147 (1993).

[3] J. Hopwood, C. R. Guarnieri, S. J. WHitehair, and J. J. Cuomo, *J. Vac. Sci. Technol. A*, **11**, 152 (1993).

[4] J. Hopwood, *Appl. Phys. Lett.*, **62**, 940 (1993).

[5] J. A. O'Neill, M. S. Barnes, and J. H. Keller, *J. Appl. Phys.*, **73**, 1621 (1993).

[6] R. Patrick, P. Schoenborn, H. Toda, and F. Bose, *J. Vac. Sci. Technol. A*, **11**, 2487 (1993).

[7] J. H. Keller, J. C. Forster, and M. S. Barnes, *J. Vac. Sci. Technol. A*, **11**, 1296 (1993).

[8] Y. Ra and C.-H. Chen, *J. Vac. Sci. Technol. A*, **11**, 2911 (1993).

[9] V. A. Godyak, R. B. Piejak, and B. M. Alexandrovich, *Plasma Sources Sci. Technol.*, in press (1994).

[10] J. Marks, K. Collins, C. L. Yang, D. Groechel, P. Keswick, and C. Cunningham, *SPIE*, **1803**, 238 (1992).

[11] J. B. Carter, J. P. Holland, E. Peltzer, B. Richardson, E. Bogle, H. T. Nguyen, Y. Melaku, D. Gates, and M. Ben-Dor, *J. Vac. Soc. Technol. A*, **11**, 1301 (1993).

[12] W. H. Press, B. P. Flannery, S. A¿ Teukolsky, W. T. Vettering, “Numerical Recipes”, (Press Syndicate of the University of Cambridge, 1986), section 3.4, page 91.

[13] L. A. Berry, E. F. Jaeger, and J. S. Tolliver, “Power Deposition Model with Capacitive Coupling”, preprint, 1994.

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551